USACERL ADP Report P-91/28
May 1991

US Army Corps
of Engineers
Construction Engineering
Research Laboratory

Maintenance Resource Prediction Model (MRPM) Individual Facility System Programmer's Manual

by
Edgar S. Neely
Robert D. Neathammer
Bill Wang

The Maintenance Resource Prediction Model (MRPM) is a computer system designed to assist in planning and programming maintenance resources, based on the anticipated resource requirements of actual installation facilities, for prediction periods of 1 to 10 years.

This manual provides system programmers with a comprehensive description of each procedure required to learn, operate, and maintain the personal computer MRPM individual facility system.

The data base and computer system are presently used by U.S. Army Corps of Engineers (USACE) designers at district and installation levels, and by resource programmers at the USACE Headquarters, Army Major Command, and installation levels. These products may also prove useful to other Government agencies and to the private sector.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official indorsement or approval of the use of such commercial products. The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN IT IS NO LONGER NEEDED

DO NOT RETURN IT TO THE ORIGINATOR

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1991	Final

4. TITLE AND SUBTITLE

Maintenance Resource Prediction Model (MRPM) Individual Facility System Programmer's Manual

5. FUNDING NUMBERS

RDTE in 1980
REIMB 1984-1988

6. AUTHOR(S)

Edgar S. Neely, Robert D. Neathammer, and Bill Wang

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Construction Engineering Research Laboratory (USACERL)
P. O. Box 4005
Champaign, IL 61824-4005

8. PERFORMING ORGANIZATION REPORT NUMBER

ADP P-91/28

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

USA Engineering and Housing Support Center
ATTN: CESHC-FM-R
Fort Belvoir, VA 22060-5516

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Copies are available from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Maintenance Resource Prediction Model (MRPM) is a computer system designed to assist in planning and programming maintenance resources, based on the anticipated resource requirements of actual installation facilities, for prediction periods of 1 to 10 years.

This manual provides system programmers with a comprehensive description of each procedure required to learn, operate, and maintain the personal computer MRPM individual facility system.

The data base and computer system are presently used by U.S. Army Corps of Engineers (USACE) designers at district and installation levels, and by resource programmers at the USACE Headquarters, Army Major Command, and installation levels. These products may also prove useful to other Government agencies and to the private sector.

14. SUBJECT TERMS

Maintenance Resource Prediction Model (MRPM)
programming manuals

15. NUMBER OF PAGES

106

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

FOREWORD

This research was conducted for the Office, Chief of Engineers, under various RDTE and FAD funding documents. Work began under RDTE in 1980 and continued in reimbursable projects from 1984 through 1989. The technical monitor was Mr. Edmund Davis (CEHSC-FM-R).

The work was performed by the Facility Systems Division (FS), U.S. Army Construction Engineering Research Laboratory (USACERL). Assistance was provided by Mr. James Stirn and Mr. Kurt Giehler of USACERL. The primary contractor on much of the data development was the Department of Architectural Engineering, Pennsylvania State University. Dr. Michael J. O'Connor is Chief, USACERL-FS. Technical editing assistance was provided by Linda Wheatley, USACERL Information Management Office.

COL Everett R. Thomas is Commander and Director of USACERL, and Dr. L.R. Shaffer is Technical Director.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

CONTENTS

MAINTENANCE RESOURCE PREDICTION MODEL (MRPM)
INDIVIDUAL FACILITY SYSTEM PROGRAMMER'S MANUAL

1 INTRODUCTION

The primary purpose of this manual is to provide the system programmers with a comprehensive description of each procedure required to learn, operate, and maintain the personal computer MRPM individual facility system. Chapter 2 describes the most efficient method for learning the functions and organization of the MRPM system. Chapter 3 defines the program flow from subroutine to subroutine. Chapter 4 contains the description of all standard or common subroutines that must be used by all programmers when writing new code or modifying existing code. Chapter 5 describes all standard common blocks that must be used when programming. Chapter 6 contains a list of standard programming packages used in the MRPM system. Chapter 7 describes the procedures to be followed during system maintenance and operation. Chapter 8 describes the overall management procedures required to operate the system. Chapter 9 describes resources required to support the MRPM system.

2 LEARNING THE FUNCTIONS

The first and most important step is to train the maintainer as a functional user of the system. Give the new person a user's manual and access to the MRPM system on a personal computer (PC). Have the person read the manual, learn the system, and write down all questions. Do not give the person any verbal description of the system. All information should be contained in the user's manual.

Revise the user's manual as needed using the programmer's questions. If the new person had the question, it is probable others will also. This method constantly improves both user and system documentation.

The user's manual is a self-teach document. The learning process takes approximately 1 week and should be very smooth and efficient. Once the maintainer knows the functions, this programmer's manual can be used.

3 PROGRAM FLOW

This chapter presents the flow of the program by functional use. Table 1-A contains the functional use as displayed on the screen, the program named, and the files as tables accessed. Table 1-B contains the same information ordered by program name. Table 1-C contains the information ordered by file name.

Table 1-A

MRPM Files*
Ordered as Shown on the Screen

Function	Program	Data Files
INSTMENU MENU		
INSTMENU	INSTMEMU EXE	INSTMENU.TB1
MAIN MENU		
MRPM	MRPMSZ EXE	MRPMTEMP.DAT MRPMTEMP.BAT
BASIC INFORMATION		
GENERAL INFORMATION		
Organization Chart	ORGCHRT EXE	ORGFCG XDB
RMF Factors	RMF-FACT EXE	RMF-FACT XDB INSTINFO DAT
F4C Conversion Codes	F4CAMS EXE	AMSF4C XDB
Report Periods	RP-DATES EXE	INSTINFO DAT ORGFCG XDB
Unit Cost Factors Unit Cost Factors	UNC-FACT EXE	UNC-FACT XDB UNCDSC XDB
U.S. Factor Description Editor	COST-DES EXE	UNCDSC XDB
Directory Location	DRIV-SPE EXE	DRSPE XDB
PREDICTION MODELS		
Prediction Model Definitions	PMDEF EXE	PMDEF XDB INSTINFO DAT PMDEF XDB

*The listing of programs is in menu order. For each MRPM function there is a corresponding program file name and data files accessed.

Table 1-A (Cont'd)

Function	Program	Data Files
F4C Prediction Model .	PMF4C EXE	PMDEF XDB PMF4C XDB INSTINFO DAT PMF4C XDB
Component Prediction Models .	PMCOMP EXE	PMCOMP XDB PMDEF XDB INSTINFO DAT PMCOMP XDB
FACILITY RESOURCE DATA		
F4C Resource Description Table .	EDF4C EXE	DES-BTSM XDB DES-TASK XDB DES-TRWD XDB F4C-YEAR XDB INSTINFO DAT
Component Tree Table .	DES-EDT1 EXE	DES-TASK XDB DES-TRWD XDB TRWDti-- XDB TRWDti-- DAT
Basic Task Table .	DES-EDT2 EXE	DES-TASK XDB DES-TRWD XDB INSTINFO XDB TASKtigi XDB TRWDti-- XDB TRWDti-- DAT
Unit of Measure .	VALEDIT EXE	VALLIST DAT INSTINFO DAT VALLIST DAT
Trade and Costs .	TRD-EDIT EXE	TRDCOSTS DAT INSTINFO DAT TRDCOSTS DAT
Task Classification .	CLASEDIT EXE	CLASLST DAT INSTINFO DAT
Equipment and Costs .	EQC-TAB EXE	EQC-TAB XDB
Work Perfomance Methods .	WP-TOP EXE	DES-TASK XDB WP-PASS DAT TASKtigi XDB WP-DESC DAT WP-PASS DAT
Total/Partial Summary Tasks .	DES-EDT3 EXE	BTSMtdpd XDB BT-PASS DAT DES-BTSM XDB DES-TASK XDB BTSMtdpd XDB BT-PASS DAT CLASLST DAT

Table 1-A (Cont'd)

Function	Program	Data Files
		INSTINFO DAT
		TRDCOSTS DAT
		TRWDti-- DAT
		TRWDti-- XDB
		VALLIST DAT
F4C Description Editor .	F4C-EDIT EXE	F4CDESC XDB
Facility Consistency Check .	CONSJSZ EXE	CTODsecq XDB
		FACILITY XDB
		TREEsecq DAT
DATA FOR INDIVIDUAL FACILITIES		
Area Identification .	AREA_TAB EXE	AREA-TAB XDB
		INSTINFO XDB
		SUB-TAB XDB
Subinstallation Identification	SUB-TAB EXE	SUB-TAB XDB
		INSTINFO DAT
Travel Zones .	TRVEDIT EXE	TRVTIME DAT
		INSTINFO DAT
Special Condition Multipier .	SCMSEL EXE	SCMDEF DAT
		SCMID DAT
		SCMPRV DAT
		INSTINFO DAT
Financial Management .	FINMANG EXE	AMSCOD XDB
		APRCOD XDB
		STDREP XDB
Facility Funding Profile .	FACFNPRF EXE	APRCOD XDB
		FFPROF XDB
		INSTINFO DAT
FACILITY INFORMATION		
Resource Calculation .	FA-CALZ EXE	AMSF4C XDB
		BTSMtdpd XDB
		CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		INSTINFO DAT
		RMF-FACT XDB
		RSMTTOTL XDB
		RSMYsecq XDB
		SCMIDid DAT
		TASKtigi XDB
		TRAVTIME DAT
		TRDCOSTS DAT
		TREEsecq DAT
	(FAMENU sub)	
	(FAOPEN sub)	
	(FAOPN1 sub)	
	(IFSCAL sub)	

Table 1-A (Cont'd)

Function	Program	Data Files
	(OCECAL sub)	
	(SQFCAL sub)	
	(STDCAL sub)	
	(TSKCAL sub)	
	(UNCCAL sub)	
	(RSSUBS sub)	
Facility Component Quantity .	CTXLOAD EXE	CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		TREEsecq DAT
		TRWDti-- DAT
		TRWDti-- XDB
	(TREEGEN EXE)	TASKaagg XDB
General Information .	FA-XEDIZ EXE	AMSF4C XDB
		AREA-TAB XDB
		F4CDESC XDB
		FACILITY XDB
		FFPROF XDB
		INSTINFO DAT
		SCMDEF XDB
		SUB-TAB XDB
		WP-DESC DAT
Query .	QUERYZ EXE	F4C-YEAR XDB
		FACILITY XDB
		RSMYsecq XDB
		TRWDti-- DAT
		TRWDti-- XDB
Copy Facility .	COPY-FAZ EXE	FACILITY XDB
Global Change to Components .	CHNG-FAZ EXE	CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		INSTINFO DAT
		TASKtigi XDB
		TREEsecq DAT
		TRWDti-- DAT
		TRWDti-- XDB
Fac Group to Dwelling Unit .	DIVIDEZ EXE	CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		TASKtigi XDB
		TREEsecq XDB
Delete Resource Files .	RSMYDELZ EXE	RSMYsecq XDB
Move to New Directory .	MOVFACZ EXE	CTODsecq XDB
		FACILITY XDB
		TREEsecq DAT
Delete Resource Total File .	DRESTO EXE	RSMTTOTL XDB

Table 1-A (Cont'd)

Function	Program	Data Files
DISPLAY RESOURCES		
Display Resources Graphically.	RSDPY0Z EXE	F4C-YEAR XDB
		FACILITY XDB
		INSTINFO DAT
		RSMYsecq XDB
		SYLCHART ASC
		TEMPCHRT
		TRWDti-- DAT
		TRWDti-- XDB
	(CHART EXE)	TEMPCHRT
Display - Financial Data .	RSDPY3 EXE	F4C-YEAR XDB
		INSTINFO DAT
		RSMYsecq XDB
		ROiiddcc XDB
		SYLCHART ASC
		TEMPCHRT
		TRWDti-- DAT
		TRWDti-- XDB
	(CHART EXE)	TEMPCHRT
Display Facility Totals .	RSMTDP EXE	FACILITY XDB
		INSTINFO DAT
		RSMTTOTL XDB
	(RSMTREPT EXE)	
FACILITY REPORTS		
F4C/AMS Organizational Summary	NEWFCG EXE	AREA-TAB XDB
		FACILITY XDB
		INSTINFO DAT
		RSMTTOTL XDB
Task Cost Report .	TSK-COSZ EXE	F4C-YEAR XDB
		FACILTIY XDB
		INSTINFO DAT
		RSMYsecq XDB
		TRWDti-- DAT
		TRWDti-- XDB
Facility Component/Quantities	TREELSTZ EXE	CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		TASKtigi XDB
		TREEsecq DAT
Orderly Yearly Task Report .	TASKREPZ EXE	AMSCOD XDB
		APRCOD XDB
		CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		INSTINFO DAT
		ORGFGC XDB

Table 1-A (Cont'd)

Function	Program	Data Files
		RSMYsecq XDB
		TREEsecq XDB
		TRWDti-- DAT
		TRWDti-- XDB
Funding Report .	REPORTZ EXE	AMSCOD XDB
		APRCOD XDB
		F4C-YEAR XDB
		INSTINFO DAT
		ORGFGC XDB
		ROiiddcc XDB
		RSMYsecq XDB
		STDREP XDB
		SUB-TAB XDB
		TRWDti-- DAT
		TRWDti-- XDB
Resource Summary .	RS-PRINZ EXE	F4C-YEAR XDB
		FACILITY XDB
		INSTINFO DAT
		RSMYsecq XDB
		TRWDti-- DAT
		TRWDti-- XDB
Copy Financial Report Files .	COPY-ROZ EXE	ORGFGC XDB
Facility Totals Report .	RSMTREPT EXE	FACILITY XDB
		INSTINFO DAT
		RSMTTOTL XDB
Combine Funding Reports .	COM-RO EXE	INSTINFO DAT
		ROz101dm XDB
Trade Index Report .	TRADEREP EXE	INSTINFO DAT
		FACILITY XDB
		RSMYsecq XDB
Query .	QUERYZ EXE	F4C-YEAR XDB
		FACILITY XDB
		RSMYsecq XDB
		TRWDti-- DAT
		TRWDti-- XDB
MODEL FACILITY		
From IFS Database		
IFS From USA Database .	IFSFAC EXE	FACILITY XDB
		F4CBLD XDB
IFS From Germany Database .	DATFAC EXE	FACILITY XDB
IFS Non Army Organization .	NOAFAC EXE	FACILITY XDB
From Input File .	FGG_CALL EXE	
	(TREEGEN EXE)	TASKWIDE XDB
		TREEWIDE DAT
		TREEWIDE XDB
		CTODsecq XDB

Table 1-A (Cont'd)

Function	Program	Data Files
		FGCQ DAT TREEsecq XDB
At Terminal .	CTXLOAD EXE (TREEGEN EXE)	CTODsecq XDB F4C-YEAR XDB FACILITY XDB TREEsecq DAT TRWDti-- DAT TRWDti-- XDB TASKtigi XDB
Input From EMS .	RSMYLOAZ EXE	FACILITY XDB INSTINFO DAT RSMYsecq XDB
Copy Facility .	COPY-FAZ EXE	FACILITY XDB CTODsecq XDB TREEsecq DAT
Global Change to Components .	CHNG-FAZ EXE (TREEGEN EXE)	CTODsecq XDB F4C-YEAR XDB FACILITY XDB INSTINFO DAT TASKtigi XDB TREEsecq DAT TRWDti-- DAT TRWDti-- XDB
Fac Group to Dwelling Unit .	DIVIDEZ EXE	CTODsecq XDB F4C-YEAR XDB FACILITY XDB TASKtigi XDB
Delete Resource Files .	RSMYDELZ EXE	RSMYsecq XDB
Move to New Directory .	MOVFACZ EXE	CTODsecq XDB RSMYsecq XDB FACILITY XDB TREEsecq DAT
Delete Resource Total File .	DRESTO EXE	RSMTTOTL XDB
REVIEW AND APPROVAL		
Generate Resource Summary .	REPRT1 EXE	AMSCOD XDB APRCOD XDB INSTINFO DAT ORGFGC XDB
Display Resource Summary .	RSDPY3 EXE	INSTINFO DAT ROiiddcc XDB SYLCHART ASC TEMPCHRT TRWDti-- DAT

Table 1-A (Cont'd)

Function	Program	Data Files
		TRWDti-- XDB
		AMSCOD XDB
		APRCOD XDB
		ORGFGC XDB
	(CHART EXE)	TEMPCHRT
Report for Total & Components .	TOTCOMP EXE	AMSCOD XDB
		APRCOD XDB
		INSTINFO DAT
		ORGFGC XDB
Ordered Component & Task Reprt .	TASKREPZ EXE	CTODsecq XDB
		APRCOD XDB
		FACILITY XDB
		INSTINFO DAT
		ORGFGC XDB
		RSMYsecq XDB
		TRWDti-- DAT
		TRWDti-- XDB
		ROiiddcc XDB
		AMSCOD XDB
		F4C-YEAR XDB
		SORTFIL DAT
Copy Financial Reports .	COPY-ROZ EXE	ORGFGC XDB
RESEARCH		
Total Analysis .	TOTCALZ EXE	FCGCALCD DAT
		FCGCALCX DAT
		INSTINFO DAT
		AMSF4C XDB
		BTSMttgg XDB
		CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		INSTINFO DAT
		RSXXTOTL XDB
		RSXXsecq XDB
		SCMIDgg DAT
		TASKtigi XDB
		TRAVTIME DAT
		TREEsecq DAT
		TRDCOSTS DAT
		RMF-FACT XDB
Display Resources .	RSDPY3 EXE	INSTINFO DAT
		ROiiddcc XDB
		SYLCHART ASC
		TEMPCHRT
		TRWDti-- DAT
		TRWDti-- XDB
		ORGFGC XDB
		APRCOD XDB
		AMSCOD XDB
	(CHART EXE)	TEMPCHRT
Print File .	PRINTFIL EXE	

Table 1-A (Cont'd)

Function	Program	Data Files
Update Task Frequencies .	BT-UPDT EXE	
Display Resource Summary .	RSDPY120 EXE	F4C-YEAR XDB INSTINFO DAT RSXXaagg XDB FACILITY XDB TRWDti-- DAT TRWDti-- XDB
CONVERSION PROGRAMS TO ASCII		
Component Tree .	TRE-UNLZ EXE	FACILITY XDB TREEsecq DAT
Component/Tasks .	CT-UNLDZ EXE	CTODsecq XDB FACILITY XDB
Basic Tasks .	BT-UNLD EXE	
10 Years Fac. Resource Summary	RS-UNLD EXE	
80 Years Fac. Resource Summary	R80-UNLZ EXE	FACILITY XDB RSXXsecq XDB
Graph 80 Year Summary .	GRAPH80 EXE	GRAPH80 ALL GRAPH80 NEW GRAPH80 AVG
	(CHART EXE)	GRAPH80 NEW
Reset Last Performed to blank .	RESETZ EXE	CTODsecq XDB FACILITY XDB INSTINFO DAT
Display 80 Year Total Summary .	RS80DP EXE	INSTINFO DAT TRWDti-- DAT TRWDti-- XDB
Graph 120 Year Summary .	GRAPH120 EXE	GRAPH120 AVG GRAPH120 ALL GRAPH120 AMM GRAPH120 NEW
	(CHART EXE)	GRAPH120 NEW
Display 120 Year Total Summary .	RS120DP EXE	INSTINFO DAT TRWDti-- DAT TRWDti-- XDB
Sub. High Cost Task From Total .	SUBTRACT EXE	
Change Facility Subdirectories .	GFI0SUB EXE	FACILITY XDB
SPECIAL PROGRAMS		
Build RSMT Total File .	RSMTBUIZ EXE	FACILITY XDB RSMTTOTL XDB

Table 1-A (Cont'd)

Function	Program	Data Files
		RSMYsecq XDB
New Facility ID's in GFI Table .	REN-FACZ EXE	FACILITY XDB
Add P & T to Facility ID .	PRE-FACZ EXE	FACILITY XDB
Rebuild Facility.xdb From Subs .	REC-FACZ EXE	FACILITY XDB
Load RMF Factors from IFS .	GRABRMF EXE	
IFS RMFS to BTRIEVE File .	INSTRMF EXE	RMF-FACT XDB
Component Index Editor .	CIDXED EXE	
Update/Rebld. Fac. Data Files .	CLAENUP EXE	INSTINFO DAT TRWDaa-- DAT TREEffff DAT F4C-YEAR XDB FACILITY XDB TASKaagg XDB CTODffff XDB TRWDaa-- XDB
	(TREEGEN EXE)	TREE XDB
Comp. U.C.F. be. USA and Germ. .	UNC-COMP EXE	COPINP DAT UNC-FACT XDB
Rebld. Sub. Fac. Data File .	RESUBFA EXE	FACILITY XDB
Delete Facility by F4C Range .	SWEEPF4C EXE	FACILITY XDB

Table 1-B

MRPM Files*
Ordered by Program Name

Program	Function	Data Files
AREA_TAB EXE	Area Identification	AREA-TAB XDB INSTINFO XDB SUB-TAB XDB
BT-UNLD EXE	Basic Tasks	
BT-UPDT EXE	Update Task Frequencies	
CHNG-FAZ EXE	Global Change to Components	CTODsecq XDB F4C-YEAR XDB FACILITY XDB INSTINFO DAT TASKtigi XDB TREEsecq DAT TRWDti-- DAT TRWDti-- XDB
CIDXED EXE	Component Index Editor	
CLAENUP EXE	Update/Rebld. Fac. Data Files	INSTINFO DAT TRWDaa-- DAT TREEffff DAT F4C-YEAR XDB FACILITY XDB TASKaagg XDB CTODffff XDB TRWDaa-- XDB TREE XDB
CLASEDIT EXE	Task Classification	CLASLST DAT INSTINFO DAT
COM-RO EXE	Combine Funding Reports	INSTINFO DAT ROz101dm XDB
CONSISZ EXE	Facility Consistency Check	CTODsecq XDB FACILITY XDB TREEsecq DAT
COPY-FAZ EXE	Copy Facility	FACILITY XDB
COPY-ROZ EXE	Copy Financial Report Files	ORGFGC XDB
COST-DES EXE	Unit Cost Factors Description	UNCDSC XDB
CT-UNLDZ EXE	Component/Tasks	CTODsecq XDB FACILITY XDB
CTXLOAD EXE	Facility Component Quantity	CTODsecq XDB F4C-YEAR XDB FACILITY XDB TREEsecq DAT

*The listing of programs is in alphabetical order. For each program there is a corresponding MRPM function and data files accessed.

Table 1-B (Cont'd)

Program	Function	Data Files
		TRWDti-- DAT
		TRWDti-- XDB
		TASKaagg XDB
DATFAC EXE	IFS From Germany Database	FACILITY XDB
DES-EDT1 EXE	Component Tree Table	DES-TASK XDB
		DES-TRWD XDB
		TRWDti-- XDB
		TRWDti-- DAT
DES-EDT2 EXE	Basic Task Table	DES-TASK XDB
		DES-TRWD XDB
		INSTINFO XDB
		TASKtigi XDB
		TRWDti-- XDB
		TRWDti-- DAT
DES-EDT3 EXE	Total/Partial Summary Tasks	BTSMtdpd XDB
		BT-PASS DAT
		DES-BTSM XDB
		DES-TASK XDB
		BTSMtdpd XDB
		BT-PASS DAT
		CLASLST DAT
		INSTINFO DAT
		TRDCOSTS DAT
		TRWDti-- DAT
		TRWDti-- XDB
		VALLIST DAT
DIVIDEZ EXE	Fac Group to Dwelling Unit	CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		TASKtigi XDB
		TREEsecq XDB
DRESTO EXE	Delete Resource Total File	RSMTTOTL XDB
DRIV-SPE EXE	Directory Location	DRSPE XDB
EDF4C EXE	F4C Resource Description Table	DES-BTSM XDB
		DES-TASK XDB
		DES-TRWD XDB
		F4C-YEAR XDB
		INSTINFO DAT
EQC-TAB EXE	Equipment and Costs	EQC-TAB XDB
F4C-EDIT EXE	F4C Description Editor	F4CDESC XDB
F4CAMS EXE	F4C Conversion Codes	AMSF4C XDB
FA-CALZ EXE	Resource Calculation	AMSF4C XDB
		BTSMtdpd XDB
		CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		INSTINFO DAT

Table 1-B (Cont'd)

Program	Function	Data Files
		RMF-FACT XDB
		RSMTTOTL XDB
		RSMYsecq XDB
		SCMIDid DAT
		TASKtigi XDB
		TRAVTIME DAT
		TRDCOSTS DAT
		TREEsecq DAT
FA-XEDIZ EXE	General Information	AMSF4C XDB
		AREA-TAB XDB
		F4CDESC XDB
		FACILITY XDB
		FFPROF XDB
		INSTINFO DAT
		SCMDEF XDB
		SUB-TAB XDB
		WP-DESC DAT
FACFNPRF EXE	Facility Funding Profile	APRCOD XDB
		FFPROF XDB
		INSTINFO DAT
FGG_CALL EXE	From Input File	TASKWIDE XDB
		TREEWIDE DAT
		TREEWIDE XDB
		CTODsecq XDB
		FGCQ DAT
		TREEsecq XDB
FINMANG EXE	Financial Management	AMSCOD XDB
		APRCOD XDB
		STDREP XDB
GFI0SUB EXE	Change Facility Subdirectories	FACILITY XDB
GRABRMF EXE	Load RMF Factors from IFS	
GRAPH120 EXE	Graph 120 Year Summary	GRAPH120 AVG
		GRAPH120 ALL
		GRAPH120 AMM
		GRAPH120 NEW
GRAPH80 EXE	Graph 80 Year Summary	GRAPH80 ALL
		GRAPH80 NEW
		GRAPH80 AVG
IFSFAC EXE	IFS From USA Database	FACILITY XDB
		F4CBLD XDB
INSTMEMU EXE	INSTMENU	INSTMENU.TB1
INSTRMF EXE	IFS RMFS to BTRIEVE File	RMF-FACT XDB
MOVFACZ EXE	Move to New Directory	CTODsecq XDB
		FACILITY XDB
		TREEsecq DAT

Table 1-B (Cont'd)

Program	Function	Data Files
MRPMSZ EXE	MAIN MENU	MRPMTEMP.DAT MRPMTEMP.BAT
NEWFCG EXE	F4C/AMS Organizational Summary	AREA-TAB XDB FACILITY XDB INSTINFO DAT RSMTTOTL XDB
NOAFAC EXE	IFS Non Army Organization	FACILITY XDB
ORGCHRT EXE	Organization Chart	ORGFCG XDB
PMCOMP EXE	Component Prediction Models	PMCOMP XDB PMDEF XDB INSTINFO DAT PMCOMP XDB
PMDEF EXE	Prediction Model Definitions	PMDEF XDB INSTINFO DAT PMDEF XDB
PMF4C EXE	F4C Prediction Model	PMDEF XDB PMF4C XDB INSTINFO DAT PMF4C XDB
PRE-FACZ EXE	Add P & T to Facility ID	FACILITY XDB
PRINTFIL EXE	Print File	
QUERYZ EXE	Query	F4C-YEAR XDB FACILITY XDB RSMYsecq XDB TRWDti-- DAT TRWDti-- XDB
R80-UNLZ EXE	80 Years Fac. Resource Summary	FACILITY XDB RSXXsecq XDB
REC-FACZ EXE	Rebuild Facility.xdb From Subs	FACILITY XDB
REN-FACZ EXE	New Facility ID's in GFI Table	FACILITY XDB
REPORTZ EXE	Funding Report	AMSCOD XDB APRCOD XDB F4C-YEAR XDB INSTINFO DAT ORGFGC XDB ROiiddcc XDB RSMYsecq XDB STDREP XDB SUB-TAB XDB TRWDti-- DAT TRWDti-- XDB
REPRT1 EXE	Generate Resource Summary	AMSCOD XDB APRCOD XDB INSTINFO DAT ORGFGC XDB

Table 1-B (Cont'd)

Program	Function	Data Files
RESETZ EXE	Reset Last Performed to blank	CTODsecq XDB FACILITY XDB INSTINFO DAT
RESUBFA EXE	Rebld. Sub. Fac. Data File	FACILITY XDB
RMF-FACT EXE	RMF Factors	RMF-FACT XDB
RP-DATES EXE	Report Periods	INSTINFO DAT ORGFCG XDB
RS-PRINZ EXE	Resource Summary	F4C-YEAR XDB FACILITY XDB INSTINFO DAT RSMYsecq XDB TRWDti-- DAT TRWDti-- XDB
RS-UNLD EXE	10 Years Fac. Resource Summary	
RS120DP EXE	Display 120 Year Total Summary	INSTINFO DAT TRWDti-- DAT TRWDti-- XDB
RS80DP EXE	Display 80 Year Total Summary	INSTINFO DAT TRWDti-- DAT TRWDti-- XDB
RSDPY0Z EXE	Display Resources Graphically	F4C-YEAR XDB FACILITY XDB INSTINFO DAT RSMYsecq XDB SYLCHART ASC TEMPCHRT TRWDti-- DAT TRWDti-- XDB TEMPCHRT
RSDPY120 EXE	Display Resource Summary	F4C-YEAR XDB INSTINFO DAT RSXXaagg XDB FACILITY XDB TRWDti-- DAT TRWDti-- XDB
RSDPY3 EXE	Display - Financial Data	F4C-YEAR XDB INSTINFO DAT RSMYsecq XDB ROiiddcc XDB SYLCHART ASC TEMPCHRT TRWDti-- DAT TRWDti-- XDB TEMPCHRT
RSDPY3 EXE	Display Resource Summary	INSTINFO DAT ROiiddcc XDB SYLCHART ASC TEMPCHRT

Table 1-B (Cont'd)

Program	Function	Data Files
		TRWDti-- DAT
		TRWDti-- XDB
		AMSCOD XDB
		APRCOD XDB
		ORGFGC XDB
		TEMPCHRT
RSMTBUIZ EXE	Build RSMT Total File	FACILITY XDB
		RSMTTOTL XDB
		RSMYsecq XDB
RSMTDP EXE	Display Facility Totals	FACILITY XDB
		INSTINFO DAT
		RSMTTOTL XDB
RSMTREPT EXE	Facility Totals Report	FACILITY XDB
		INSTINFO DAT
		RSMTTOTL XDB
RSMYDELZ EXE	Delete Resource Files	RSMYsecq XDB
RSMYLOAZ EXE	Input From EMS	FACILITY XDB
		INSTINFO DAT
		RSMYsecq XDB
SCMSEL EXE	Special Condition Multipier	SCMDEF DAT
		SCMID DAT
		SCMPRV DAT
		INSTINFO DAT
SUB-TAB EXE	Subinstallation Identification	SUB-TAB XDB
		INSTINFO DAT
SUBTRACT EXE	Sub. High Cost Task From Total	
SWEEPF4C EXE	Delete Facility by F4C Range	FACILITY XDB
TASKREPZ EXE	Orderly Yearly Task Report	AMSCOD XDB
		APRCOD XDB
		CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		INSTINFO DAT
		ORGFGC XDB
		RSMYsecq XDB
		TREEsecq XDB
		TRWDti-- DAT
		TRWDti-- XDB
TOTCALZ EXE	Total Analysis	FCGCALCD DAT
		FCGCALCX DAT
		INSTINFO DAT
		AMSF4C XDB
		BTSMttgg XDB
		CTODsecq XDB
		F4C-YEAR XDB
		FACILITY XDB
		INSTINFO DAT
		RSXXTOTL XDB

Table 1-B (Cont'd)

Program	Function	Data Files
		RSXXsecq XDB SCMIDgg DAT TASKtigi XDB TRAVTIME DAT TREEsecq DAT TRDCOSTS DAT RMF-FACT XDB
TOTCOMP EXE	Report for Total & Components	AMSCOD XDB APRCOD XDB INSTINFO DAT ORGFGC XDB
TRADEREP EXE	Trade Index Report	INSTINFO DAT FACILITY XDB RSMYsecq XDB
TRD-EDIT EXE	Trade and Costs	TRDCOSTS DAT INSTINFO DAT TRDCOSTS DAT
TRE-UNLZ EXE	Component Tree	FACILITY XDB TREEsecq DAT
TREEGEN EXE	Generate Tree file	TREE.XDB TREE.DAT
TREELSTZ EXE	Facility Component/Quantities	CTODsecq XDB F4C-YEAR XDB FACILITY XDB TASKtigi XDB TREEsecq DAT
TRVEDIT EXE	Travel Zones	TRVTIME DAT INSTINFO DAT
TSK-COSZ EXE	Task Cost Report	F4C-YEAR XDB FACILTIY XDB INSTINFO DAT RSMYsecq XDB TRWDti-- DAT TRWDti-- XDB
UNC-COMP EXE	Comp. U.C.F. be. USA and Germ.	COPINP DAT UNC-FACT XDB
UNC-FACT EXE	Unit Cost Factors	UNC-FACT XDB UNCDSC XDB
VALEDIT EXE	Unit of Measure	VALLIST DAT INSTINFO DAT VALLIST DAT
WP-TOP EXE	Work Perfomance Methods	DES-TASK XDB WP-PASS DAT TASKtigi XDB WP-DESC DAT WP-PASS DAT

Table 1-C

MRPM Files*
Ordered by Data File Name

Data File	Program
AMSF4C XDB	F4CAMS EXE
	FA-CALZ EXE
	FA-XEDIZ EXE
	TOTCALZ EXE
APRCOD XDB	FACFNPRF EXE
	FINMANG EXE
	REPORTZ EXE
	REPRT1 EXE
	RSDPY3 EXE
	TASKREPZ EXE
	TOTCOMP EXE
AREA-TAB XDB	AREA_TAB EXE
	FA-XEDIZ EXE
	NEWFCG EXE
BTSMtdpd XDB	DES-EDT3 EXE
	FA-CALZ EXE
	TOTCALZ EXE
BT-PASS DAT	DES-EDT3 EXE
CLASLST DAT	CLASEDIT EXE
	DES-EDT3 EXE
COPINP DAT	UNC-COMP EXE
CTODsecq XDB	CLAENUP EXE
	CHNG-FAZ EXE
	CONSISZ EXE
	COPY-FAZ EXE
	CTXLOAD EXE
	CT-UNLDZ EXE
	DIVIDEZ EXE
	FA-CALZ EXE
	MOVFACZ EXE
	RESETZ EXE
	TASKREPZ EXE
	TOTCALZ EXE
	TREEGEN EXE
	TREELSTZ EXE
DES-BTSM XDB	DES-EDT3 EXE
	EDF4C EXE
DES-TASK XDB	DES-EDT1 EXE
	DES-EDT2 EXE
	DES-EDT3 EXE
	EDF4C EXE
	WP-TOP EXE

*The listing of data file name is in alphabetical order. Each data file has corresponding execut files.

Table 1-C (Cont'd)

Data File	Program
DES-TRWD XDB	DES-EDT1 EXE
	DES-EDT2 EXE
	EDF4C EXE
RSPE XDB	DRIV-SPE EXE
EQC-TAB XDB	EQC-TAB EXE
F4CBLD XDB	IFSFAC EXE
F4CDESC XDB	F4C-EDIT EXE
	FA-XEDIZ EXE
F4C-YEAR XDB	CHNG-FAZ EXE
	CLAENUP EXE
	CTXLOAD EXE
	DIVIDEZ EXE
	EDF4C EXE
	FA-CALZ EXE
	QUERYZ EXE
	REPORTZ EXE
	RSDPY0Z EXE
	RSDPY120 EXE
	RSDPY3 EXE
	RS-PRINZ EXE
	TASKREPZ EXE
	TOTCALZ EXE
	TREEGEN EXE
	TREELSTZ EXE
	TSK-COSZ EXE
FACILITY XDB	CHNG-FAZ EXE
	CLAENUP EXE
	CONSISZ EXE
	COPY-FAZ EXE
	CTXLOAD EXE
	CT-UNLDZ EXE
	DATFAC EXE
	DIVIDEZ EXE
	FA-CALZ EXE
	FA-XEDIZ EXE
	GFI0SUB EXE
	IFSFAC EXE
	MOVFACZ EXE
	NEWFCG EXE
	NOAFAC EXE
	PRE-FACZ EXE
	QUERYZ EXE
	R80-UNLZ EXE
	REC-FACZ EXE
	REN-FACZ EXE
	RESETZ EXE
	RESUBFA EXE
	RSDPY0Z EXE
	RSDPY120 EXE
	RSMTBUIZ EXE
	RSMTDP EXE
	RSMTREPT EXE
	RSMYLOAZ EXE

Table 1-C (Cont'd)

Data File		Program
		RS-PRINZ EXE
		SWEEPF4C EXE
		TASKREPZ EXE
		TOTCALZ EXE
		TRADEREP EXE
		TREEGEN EXE
		TREELSTZ EXE
		TRE-UNLZ EXE
		TSK-COSZ EXE
FCGCALCi	DAT	TOTCALZ EXE
FFPROF	XDB	FACFNPRF EXE
		FA-XEDIZ EXE
FGCQ	DAT	TREEGEN EXE
GRAPH120	ALL	GRAPH120 EXE
GRAPH120	AMM	GRAPH120 EXE
GRAPH120	AVG	GRAPH120 EXE
GRAPH120	NEW	CHART EXE
		GRAPH120 EXE
GRAPH80	ALL	GRAPH80 EXE
GRAPH80	AVG	GRAPH80 EXE
GRAPH80	NEW	CHART EXE
		GRAPH80 EXE
INSTINFO	DAT	CHNG-FAZ EXE
		CLAENUP EXE
		CLASEDIT EXE
		COM-RO EXE
		DES-EDT3 EXE
		EDF4C EXE
		FACFNPRF EXE
		FA-CALZ EXE
		FA-XEDIZ EXE
		NEWFCG EXE
		PMCOMP EXE
		PMDEF EXE
		PMF4C EXE
		REPORTZ EXE
		REPRT1 EXE
		RESETZ EXE
		RMF-FACT EXE
		RP-DATES EXE
		RS120DP EXE
		RS80DP EXE
		RSDPY0Z EXE
		RSDPY120 EXE
		RSDPY3 EXE
		RSDPY3 EXE
		RSMTDP EXE
		RSMTREPT EXE

Table 1-C (Cont'd)

Data File	Program
	RSMYLOAZ EXE
	RS-PRINZ EXE
	SCMSEL EXE
	SUB-TAB EXE
	TASKREPZ EXE
	TOTCALZ EXE
	TOTCOMP EXE
	TRADEREP EXE
	TRD-EDIT EXE
	TREEGEN EXE
	TRVEDIT EXE
	TSK-COSZ EXE
	VALEDIT EXE
	AREA_TAB EXE
	DES-EDT2 EXE
INSTMENU TB1	INSTMEMU EXE
MRPMTEMP BAT	MRPMSZ EXE
MRPMTEMP DAT	MRPMSZ EXE
ORGFGC XDB	COPY-ROZ EXE
	ORGCHRT EXE
	REPORTZ EXE
	REPRT1 EXE
	RP-DATES EXE
	RSDPY3 EXE
	TASKREPZ EXE
	TOTCOMP EXE
PMCOMP XDB	PMCOMP EXE
PMDEF XDB	PMCOMP EXE
	PMDEF EXE
	PMF4C EXE
PMF4C XDB	PMF4C EXE
RMF-FACT XDB	FA-CALZ EXE
	INSTRMF EXE
	RMF-FACT EXE
	TOTCALZ EXE
ROiiddcc XDB	REPORTZ EXE
	RSDPY3 EXE
	TASKREPZ EXE
	COM-RO EXE
RSMTTOTL XDB	DRESTO EXE
	FA-CALZ EXE
	NEWFCG EXE
	RSMTBUIZ EXE
	RSMTDP EXE
	RSMTREPT EXE
RSMYsecq XDB	FA-CALZ EXE
	MOVFACZ EXE
	QUERYZ EXE

Table 1-C (Cont'd)

Data File	Program
	REPORTZ EXE
	RSDPY0Z EXE
	RSDPY3 EXE
	RSMTBUIZ EXE
	RSMYDELZ EXE
	RSMYLOAZ EXE
	RS-PRINZ EXE
	TASKREPZ EXE
	TRADEREP EXE
	TSK-COSZ EXE
RSXXaagg XDB	RSDPY120 EXE
	R80-UNLZ EXE
	TOTCALZ EXE
RSXXTOTL XDB	TOTCALZ EXE
SCMDEF DAT	SCMSEL EXE
	FA-XEDIZ EXE
SCMID DAT	SCMSEL EXE
SCMIDgg DAT	TOTCALZ EXE
SCMIDid DAT	FA-CALZ EXE
SCMPRV DAT	SCMSEL EXE
SORTFIL DAT	TASKREPZ EXE
STDREP XDB	FINMANG EXE
	REPORTZ EXE
SUB-TAB XDB	AREA_TAB EXE
	FA-XEDIZ EXE
	REPORTZ EXE
	SUB-TAB EXE
SYLCHART ASC	RSDPY0Z EXE
	RSDPY3 EXE
TASKaagg XDB	CLAENUP EXE
	TREEGEN EXE
TASKtigi XDB	CHNG-FAZ EXE
	DES-EDT2 EXE
	DIVIDEZ EXE
	FA-CALZ EXE
	TOTCALZ EXE
	TREEGEN EXE
	TREELSTZ EXE
	WP-TOP EXE
TASKWIDE XDB	TREEGEN EXE
TEMPCHRT	CHART EXE
	RSDPY0Z EXE
	RSDPY3 EXE

Table 1-C (Cont'd)

Data File	Program
TRAVTIME DAT	FA-CALZ EXE
	TOTCALZ EXE
TRDCOSTS DAT	DES-EDT3 EXE
	FA-CALZ EXE
	TOTCALZ EXE
	TRD-EDIT EXE
TREE XDB	TREEGEN EXE
TREEffff DAT	CLAENUP EXE
	CHNG-FAZ EXE
	CONSISZ EXE
	COPY-FAZ EXE
	CTXLOAD EXE
	FA-CALZ EXE
	MOVFACZ EXE
	TOTCALZ EXE
	TREEGEN EXE
	TREELSTZ EXE
	TRE-UNLZ EXE
	DIVIDEZ EXE
	TASKREPZ EXE
	TREEGEN EXE
TREEWIDE DAT	TREEGEN EXE
TREEWIDE XDB	TREEGEN EXE
TRVTIME DAT	TRVEDIT EXE
TRWDaa-- DAT	CLAENUP EXE
TRWDaa-- XDB	CLAENUP EXE
TRWDti-- DAT	CHNG-FAZ EXE
	CTXLOAD EXE
	DES-EDT1 EXE
	DES-EDT2 EXE
	DES-EDT3 EXE
	QUERYZ EXE
	REPORTZ EXE
	RS120DP EXE
	RS80DP EXE
	RSDPY0Z EXE
	RSDPY120 EXE
	RSDPY3 EXE
	RS-PRINZ EXE
	TASKREPZ EXE
	TREEGEN EXE
	TSK-COSZ EXE
TRWDti-- XDB	CHNG-FAZ EXE
	CTXLOAD EXE
	DES-EDT1 EXE
	DES-EDT2 EXE
	DES-EDT3 EXE
	QUERYZ EXE
	REPORTZ EXE

Table 1-C (Cont'd)

Data File		Program
		RS120DP EXE
		RS80DP EXE
		RSDPY0Z EXE
		RSDPY120 EXE
		RSDPY3 EXE
		RS-PRINZ EXE
		TASKREPZ EXE
		TREEGEN EXE
		TSK-COSZ EXE
UNCDSC	XDB	COST-DES EXE
		UNC-FACT EXE
UNC-FACT	XDB	UNC-COMP EXE
		UNC-FACT EXE
VALLIST	DAT	DES-EDT3 EXE
		VALEDIT EXE
WP-DESC	DAT	FA-XEDIZ EXE
		WP-TOP EXE
WP-PASS	DAT	WP-TOP EXE

4 STANDARD SUBROUTINES

Table 2 contains an alphabetical list of all standard subroutines with descriptions. Each subroutine is defined in detail.

Table 2

Standard Subroutines

```
*************************************************************************
*
* Name        = ansist                                                  *
*                                                                       *
* Author      = Bobby Adams                                             *
*                                                                       *
* Function    = To perform various ansi-standard routines by sending the *
*               appropriate sequence of commands to the terminal.  Note, *
*               the command parameter (com) can be in upper or lower case. *
*                                                                       *
* Usage       = character com*3                                         *
*               integer*2 prm1, prm2                                    *
*               call ansist (com, prm1, prm2)                           *
*                                                                       *
* Parameters =                                                          *
*       com  - SGR (select graphic rendition)                           *
*       prm1 - 0 (turn all attributes off)                              *
*              1 (increase screen intensity)                            *
*              4 (dim screen intensity)                                 *
*              5 (blinking)                                             *
*              7 (reverse video)                                        *
*       prm2 - not used (use 0 for consistency)                         *
*                                                                       *
*       com  - HVP (horizontal and vertical position)                   *
*       prm1 - 1-24 (row in which to place cursor)                      *
*       prm2 - 1-80 (column in which to place cursor)                   *
*                                                                       *
*       com  - CUU (cursor up, doesn't change column position)          *
*       prm1 - number of rows to move up (won't move above top margin)  *
*       prm2 - not used (use 0 for consistency)                         *
*                                                                       *
*       com  - CUD (cursor down, doesn't change column position)        *
*       prm1 - number of rows to move down (won't go below bottom margin) *
*       prm2 - not used (use 0 for consistency)                         *
*                                                                       *
*       com  - CUF (cursor forward, doesn't change row position)        *
*       prm1 - number of cols to move right (won't go past right margin) *
*       prm2 - not used (use 0 for consistency)                         *
*                                                                       *
*       com  - CUB (cursor backward, doesn't change row position)       *
*       prm1 - number of cols to move left (won't go past left margin)  *
*       prm2 - not used (use 0 for consistency)                         *
*                                                                       *
*       com  - ED (erase in display, clear screen)                      *
*       prm1 - not used (use 0 for consistency)                         *
*       prm2 - not used (use 0 for consistency)                         *
*                                                                       *
*       com  - EL (erase in line)                                       *
*       prm1 - 0 (erase from cursor position to the end of line, inclusive) *
*              2 (erase the entire line)                                *
*       prm2 - line number to erase                                     *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  upper  compac                                 *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*****************************************************************************
* Name        = box (assembler routine)                                     *
*                                                                           *
* Author      = Russ Hougland                                               *
*                                                                           *
* Function    = Draws a box with an upper left corner at (x1,y1) and a lower *
*               right corner at (x2,y2).  The sides of the box are either   *
*               single lines (type = 1) or double lines (type = 2).  A box can *
*               also be erased (type = 0).                                  *
*                                                                           *
* Usage       = integer*2 x1, y1, x2, y2, type, att                         *
*               call box (x1, y1, x2, y2, type, att)                        *
*                                                                           *
* Parameters =                                                              *
*       x1    - row of the upper left hand corner                           *
*       y1    - column of the upper left hand corner                        *
*       x2    - row of the lower right hand corner (x2 must be > x1)        *
*       y2    - column of the lower right hand corner (y2 must be > y1)     *
*       type  - 0 (blank lines, erase a box)                                *
*               1 (single line box)                                         *
*               2 (double line box)                                         *
*       att   - the screen attribute to use on the box (see wt)             *
*                                                                           *
* Returns     = None.                                                       *
*                                                                           *
* Calls       = None.                                                       *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*****************************************************************************
* Name        = cd (convert date)                                           *
*                                                                           *
* Author      = Russ Hougland                                               *
*                                                                           *
* Function    = Converts a date from the internal storage format to the     *
*               format specified.                                           *
*                                                                           *
* Usage       = character fmt*01, cd*11, date*11, newdate*11                *
*               newdate = cd (date, fmt)                                    *
*                                                                           *
* Parameters =                                                              *
*       date  - the date to be converted (must be in the internal format of *
*               'YYYY/MMM/DD' i.e. '1985/001/01')                           *
*       fmt   - the date format to convert to (can be lower or upper case)  *
*               'E' (english format of 'MMM/DD/YYYY' i.e. 'JAN/01/1985')    *
*               'M' (metric format of 'DD/MMM/YYYY' i.e. '01/JAN/1985')     *
*                                                                           *
* Returns     = The converted date stored in cd.  If the date passed is empty *
*               or in error, cd returns an empty date (i.e. '   /  /    ')  *
*                                                                           *
* Calls       = fortran:  ichar                                             *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*******************************************************************************
* Name        = cdat                                                          *
*                                                                             *
* Author      = Kevin Stewart                                                 *
*                                                                             *
* Function    = Converts a date from the internal database storage format to  *
*               a real number (YYYY. + MM/12) that is to the nearest month.   *
*                                                                             *
* Usage       = real*4    num, cdat                                           *
*               character date*11                                             *
*               num = cdat (date)                                             *
*                                                                             *
* Parameters =                                                                *
*       date - the date to be converted (in internal format 'YYYY/MMM/DD')    *
*                                                                             *
* Returns     = The date as a real number (YYYY. + MM/12).                    *
*                                                                             *
* Calls       = fsc.lib:  iconv                                               *
*                                                                             *
* Commons     = None.                                                         *
*******************************************************************************
```

```
*******************************************************************************
* Name        = chkbit (assembler)                                            *
*                                                                             *
* Author      = Russ Hougland                                                 *
*                                                                             *
* Function    = Checks whether a bit is set or reset.                         *
*                                                                             *
* Usage       = integer*2 code, bit                                           *
*               logical*2 i, chkbit                                           *
*               i = chkbit (code, bit)                                        *
*                                                                             *
* Parameters =                                                                *
*       code - variable to check                                              *
*       bit  - the bit in variable (code) to check (0 - 15)                   *
*                                                                             *
* Returns     = True if the bit is set.                                       *
*                                                                             *
* Calls       = None.                                                         *
*                                                                             *
* Commons     = None.                                                         *
*******************************************************************************
```

```
*************************************************************************
* Name        = clrbox (clear box)                                      *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Clears the interior area of a box with the upper left-hand *
*               corner at (x1,y1) and the lower right-hand corner at (x2,y2). *
*               The attribute of the interior is determined by attr.    *
*                                                                       *
* Usage       = integer*2 x1, y1, x2, y2, attr                          *
*               call clrbox (x1, y1, x2, y2, attr)                      *
*                                                                       *
* Parameters =                                                          *
*       x1    - row of the upper left hand corner                       *
*       y1    - column of the upper left hand corner                    *
*       x2    - row of the lower right hand corner (x2 must be > x1)    *
*       y2    - column of the lower right hand corner (y2 must be > y1) *
*       attr  - attribute of the interior of the box (see wt)           *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  scroll                                        *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = clrmod (color mode - assembler)                         *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = To determine if the user's computer is in the color mode. *
*                                                                       *
* Usage       = logical*2 clrmod, i                                     *
*               i = clrmod ()                                           *
*                                                                       *
* Parameters = None.                                                    *
*                                                                       *
* Returns     = clrmod                                                  *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = cmp2sd (compare to system date)                         *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = To compare a date to the system date.                   *
*                                                                       *
* Usage       = character dat*11                                        *
*               integer*2 flag                                          *
*               call cmp2sd (dat, flag)                                 *
*                                                                       *
* Parameters =                                                          *
*       dat   - the date to be compared (must be in internal format i.e.*
*               'YYYY/MMM/DD')                                          *
*       flag  - indicates results of the comparison                     *
*               -1 (if the date is before the system date)              *
*                0 (if the two dates are equal)                         *
*               +1 (if the date is after the system date)               *
*                                                                       *
* Returns     = flag (indicates results of comparison).                 *
*                                                                       *
* Calls       = fsc.lib:  sysdat                                        *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = command                                                 *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Execute a DOS command from within a program.            *
*                                                                       *
* Usage       = logical*2 error                                         *
*               character cmnd*(*)                                      *
*               call command (cmnd, error) or                           *
*               call command ('dir *.for'c,error)                       *
*                                                                       *
* Parameters =                                                          *
*    error    - true if an error occurred during execution.            *
*    cmnd     - character variable containing the DOS command to execute (if
*               a variable is used, the string contained by it must be  *
*               terminated with a null).                                *
*    literal  - if the DOS command to be executed is contained in a literal
*               string, the letter c must follow immediately after the string.
*                                                                       *
* Returns     = if error is true, a message will have been written to the
*               file error.dat                                          *
*                                                                       *
* Calls       = fsc.lib:  rpterr                                        *
*               fortran:  system                                        *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = compac                                                  *
*                                                                       *
* Author      = Bobby Adams                                             *
*                                                                       *
* Function    = Moves all blank characters in a string to the end of the*
*               string and returns the number of characters in the string.*
*                                                                       *
* Usage       = integer*2 num                                           *
*               character str*(*)                                       *
*               call compac (str, num)                                  *
*                                                                       *
* Parameters =                                                          *
*         str - the string variable of any length                       *
*         num - the number of non-blank characters in the string        *
*                                                                       *
* Returns     = It returns num, the length of character string stored in str.*
*                                                                       *
* Calls       = fortran:  len                                           *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = contain (logical function)                              *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Determines if a substring (sub) is contained within another*
*               string (str).                                           *
*                                                                       *
* Usage       = logical*2 contain, i                                    *
*               character sub*(*), str*(*)                              *
*               i = contain (str, sub)                                  *
*                                                                       *
* Parameters =                                                          *
*        str  - the string to search                                    *
*        sub  - the substring to search for                             *
*                                                                       *
* Returns     = contain is true if the substring is found within the target*
*               string.                                                 *
*                                                                       *
* Calls       = fortran:  len                                           *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = csort                                                   *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = This is an interface to a sort routine (written in c by Steve *
*               Dorner and also contained in fsc.lib).  It will sort a  *
*               file using multiple keys in either ascending or descending *
*               order.                                                  *
*                                                                       *
* Usage       = integer*4 addr1, addr2, locfar                          *
*               character control*(*), filen*(*)                        *
*                                                                       *
*               addr1 = locfar (control)                                *
*               addr2 = locfar (filen)                                  *
*               call csort (addr1, addr2)                               *
*                                                                       *
* Parameters =                                                          *
*    addr1    - the address of the control string                       *
*    addr2    - the address of the name of the file to be sorted        *
*    filen    - character string containing the name of the file to be sorted. *
*               the file name must be terminated with a null.           *
*    control  - a character variable containing the command string for the *
*               sort.  the command string must be terminated with a null. *
*               the format of control is as follows:                    *
*                  if first character is 's', second character specifies *
*                  the field separator character.                       *
*                  followed by zero or more words of the form:          *
*                     [b][c|n][r][<m>][.<n>][-[<m2>][.<n2>]             *
*                     b ignore leading blanks                           *
*                     c ignore case                                     *
*                     n numeric comparison                              *
*                     r reverse order                                   *
*                     <m> field number                                  *
*                     .<n> begin column                                 *
*                     <m2> end field number                             *
*                     .<n2> end column                                  *
* Returns     = None.                                                   *
* Calls       = fsc.lib:  sort                                          *
* Commons     = None.                                                   *
*************************************************************************


*************************************************************************
* Name        = csrpos (assembler routine)                              *
*                                                                       *
* Author      = Dan Weidenfeld                                          *
*                                                                       *
* Function    = To position the cursor.                                 *
*                                                                       *
* Usage       = integer*2 row, col                                      *
*               call csrpos (row, col)                                  *
*                                                                       *
* Parameters =                                                          *
*         row - the row to position to - 1 (0-24)                       *
*         col - the column to position to - 1 (0-79)                    *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = cursor (assembler routine)                              *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Change the type of cursor used.                         *
*                                                                       *
* Usage       = integer*2 type                                          *
*               call cursor (type)                                      *
*                                                                       *
* Parameters =                                                          *
*       type - 0 (no cursor)                                            *
*              1 (underline cursor)                                     *
*              2 (dash cursor)                                          *
*              3 (overscore cursor)                                     *
*              4 (block cursor)                                         *
*              5 (bottom-half block cursor)                             *
*              6 (top-half block cursor)                                *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = dantim (assembler routine)                              *
*                                                                       *
* Author      = Dan Weidenfeld                                          *
*                                                                       *
* Function    = To get the system time.                                 *
*                                                                       *
* Usage       = integer*2 hours, min, secs                              *
*               call gettim (hours, min, secs)                          *
*                                                                       *
* Parameters =                                                          *
*      hours - the hour of the day in military form (i.e. 1 pm is 13)   *
*      min   - the minutes                                              *
*      secs  - the seconds                                              *
*                                                                       *
* Returns     = the time of day.                                        *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*****************************************************************************
* Name        = daydif (integer function)                                   *
*                                                                           *
* Author      = Russ Hougland                                               *
*                                                                           *
* Function    = Returns the number of days between two dates.               *
*                                                                           *
* Usage       = character sdate*11, edate*11                                *
*               integer*4 days, daydif                                      *
*               days = daydif (sdate, edate)                                *
*                                                                           *
* Parameters =                                                              *
*      sdate - the starting date (in the internal format)                   *
*      edate - the ending date (in the internal format)                     *
*                                                                           *
* Returns     = The number of days between the two dates.  If an invalid date *
*               is passed, the function returns the value -9,999.           *
*                                                                           *
* Calls       = fsc.lib:  iconv                                             *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
*****************************************************************************
* Name        = delay                                                       *
*                                                                           *
* Author      = Russ Hougland                                               *
*                                                                           *
* Function    = Delays program action for a specified number of seconds.    *
*                                                                           *
* Usage       = integer*2 secs                                              *
*               call delay (secs)                                           *
*                                                                           *
* Parameters =                                                              *
*       secs - the number of seconds to delay                               *
*                                                                           *
* Returns     = None.                                                       *
*                                                                           *
* Calls       = fortran:  gettim                                            *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
*****************************************************************************
* Name        = dskcnf (assembler routine)                                  *
*                                                                           *
* Author      = Paul Shih                                                   *
*                                                                           *
* Function    = find out the disks type                                     *
*                                                                           *
* Usage       = integer*2 info(26)                                          *
*               call dskcnf(info)                                           *
*                                                                           *
* Returns     =                                                             *
*       Entry 1 is drive A                                                  *
*             2 is drive B and so on .                                      *
*                                                                           *
* Commons     =                                                             *
*             INVALID_DSK          0                                        *
*             REMOTE_DSK          -1                                        *
*             LOCAL_RAM_DSK        1                                        *
*             LOCAL_FIXED_DSK      2                                        *
*             LOCAL_FLOPPY_DSK     3                                        *
*                                                                           *
*****************************************************************************
```

```
*****************************************************************************
* Name        = erase                                                       *
*                                                                           *
* Author      = Russ Hougland                                               *
*                                                                           *
* Function    = Erases the lines between two rows (line1 and line2),        *
*               inclusive.                                                  *
* Usage       = integer*2 line1, line2, att                                 *
*               call erase (line1, line2, att)                              *
*                                                                           *
* Parameters =                                                              *
*      line1 - the erasing starts with this line (line1 must be <= line2)   *
*      line2 - the erasing ends with this line (line2 must be >= line1)     *
*      att   - the attribute to fill the erased field with                  *
*                                                                           *
* Returns     = None.                                                       *
*                                                                           *
* Calls       = fsc.lib:  scroll                                            *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*****************************************************************************
* Name        = execute                                                     *
*                                                                           *
* Author      = Russ Hougland                                               *
*                                                                           *
* Function    = Suspends the program currently running and activates the    *
*               new program (filnam).  When the new program (filnam)        *
*               terminates, the original program is awakened and resumes its*
*               execution with the next instruction after the call to this  *
*               subroutine.                                                 *
*                                                                           *
* Usage       = logical*2 error                                             *
*               character filnam*(*)                                        *
*               call execute (filnam, error) or                             *
*               call execute ('prog2.exe'c, error)                          *
*                                                                           *
* Parameters =                                                              *
*    error    - true if an error occurred during execution                  *
*    filnam   - character variable containing the program name to execute (if *
*               a variable is used, the string contained by it must be      *
*               terminated with a null).                                    *
*    literal  - if the program name to be executed is contained in a literal *
*               string, the letter c must follow immediately after the string. *
*                                                                           *
* Returns     = if error is true, a message will have been written to the   *
*               file error.dat                                              *
*                                                                           *
* Calls       = fortran:  spawnlp                                           *
*               fsc.lib:  rpterr                                            *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*************************************************************************
* Name        = fkeys                                                   *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Turns on/off the function display fields on line 25 (should *
*               be used in conjunction with fline).                     *
*                                                                       *
* Usage       = integer*2 f1, f2, f3, f4, f5, f6, f7, f8, f9, f10       *
*               call fkeys (f1, f2, f3, f4, f5, f6, f7, f8, f9, f10)    *
*                                                                       *
* Parameters =                                                          *
*   f1 - f10 - 0 (no change)                                            *
*              1 (turn display field off)                               *
*              2 (turn display field on) and displays (by default):     *
*                f1  - Help                                             *
*                f2  - Keys                                             *
*                f3  - Add                                              *
*                f4  - Delete                                           *
*                f5  - Edit                                             *
*                f6  - Find                                             *
*                f7  - List                                             *
*                f8  -                                                  *
*                f9  - Next                                             *
*                f10 - Exit                                             *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  wt                                            *
*                                                                       *
* Commons     = ffldcom                                                 *
*                 fflds (array 10x2 of character*6).  if needed, the field to *
*                 be displayed can be changed by modifying the contents of *
*                 fflds (i.e. fflds(function key,2) = 'Change').        *
*************************************************************************
```

```
*************************************************************************
* Name        = fline                                                   *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Initially writes the function keys' display line.       *
*                                                                       *
* Usage       = call fline                                              *
*                                                                       *
* Parameters = None.                                                    *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  wt                                            *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = frames (assembler routine)                              *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = To save up to five video screen windows into memory for later *
*               restoration.  When saving a window, the row and column  *
*               parameters specify the area to be saved.  When restoring a *
*               window, the row and column parameters specify the location the *
*               window should be restored to.                           *
*                                                                       *
* Usage       = integer*2 action, frame, r1, c1, r2, c2                 *
*               call frames (action, frame, r1, c1, r2, c2)             *
*                                                                       *
* Parameters =                                                          *
*      action - the action to take                                      *
*               1 (save the window to memory)                           *
*               anything else (restore the window from memory)          *
*      frame  - memory frame to save/restore the window to/from         *
*               1 - 5 (possible frames to use)                          *
*      r1     - row of the upper left hand corner of the window  (1-25) *
*      c1     - column of the upper left hand corner of the window (1-80) *
*      r2     - row of the lower right hand corner of the window (1-25) *
*      c2     - column of the lower right hand corner of the window (1-80) *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = gtinsk (get instruction key)                            *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Returns the ASCII integer value for the non-alphanumeric keys *
*               (e.g. function keys, Esc, Ins, etc.).  Alphanumeric keys are *
*               ignored.  Those keys that send two codes (i.e. a nul and then *
*               another number) will only return to the calling program the *
*               second number.                                          *
*                                                                       *
* Usage       = integer*2 key                                           *
*               call gtinsk (key)                                       *
*                                                                       *
* Parameters =                                                          *
*         key = ASCII integer value of the depressed key                *
*                                                                       *
* Returns     = key (the ASCII integer value of the depressed non-alphanumeric *
*               key)                                                    *
*                                                                       *
* Calls       = fsc.lib:  inkey                                         *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*****************************************************************************
* Name        = iconv (convert character string to integer)                 *
*                                                                           *
* Author      = Bobby Adams                                                 *
*                                                                           *
* Function    = To convert a character string of numbers to its integer     *
*               equivalent.                                                 *
*                                                                           *
* Usage       = character str*(*)                                           *
*               integer*4 num, iconv                                        *
*               logical*2 err                                               *
*               num = iconv (str, err)                                      *
*                                                                           *
* Parameters =                                                              *
*         str - the character string containing the number to convert       *
*         err - true if an error occurrs                                    *
*                                                                           *
* Returns     = iconv and err (if err, then iconv equals 0)                 *
*                                                                           *
* Calls       = fortran:  ichar                                             *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************

*****************************************************************************
* Name        = inkey (assembler routine)                                   *
*                                                                           *
* Author      = Dan Weidenfeld                                              *
*                                                                           *
* Function    = To perform a single character read from the keyboard.       *
*                                                                           *
* Usage       = integer*2 i, inkey, echo                                    *
*               i = inkey (echo)                                            *
*                                                                           *
* Parameters =                                                              *
*        echo - if equal to zero the character is not echoed to the screen. *
*                                                                           *
* Returns     = the ascii value of the character entered from the keyboard. *
*                                                                           *
* Calls       = None.                                                       *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************

*****************************************************************************
* Name        = inout (assembler routine)                                   *
*                                                                           *
* Author      = Dan Weidenfeld                                              *
*                                                                           *
* Function    = To perform a single character write to the screen.          *
*                                                                           *
* Usage       = character char*01                                           *
*               call inout (char)                                           *
*                                                                           *
* Parameters =                                                              *
*        char - the character to be displayed on the screen                 *
*                                                                           *
* Returns     = None.                                                       *
*                                                                           *
* Calls       = None.                                                       *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*****************************************************************************
* Name        = isfile (assembler routine)                                  *
*               integer*2 function isfile                                   *
*                                                                           *
* Author      = Paul Shih                                                   *
*                                                                           *
* Function    = To check file or subdirectory status                        *
*                                                                           *
* Usage       = integer*2 i                                                 *
*               i = isfile('test.dat'c)                                     *
*                                                                           *
* Returns     = -1 : if file not found                                      *
*       Bit 0 on   : Read-Only file                                         *
*       Bit 1 on   : Hidden file                                            *
*       Bit 2 on   : System file                                            *
*       Bit 3 on   : Read-Only file                                         *
*       Bit 4 on   : Read-Only file                                         *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*****************************************************************************
* Name        = line (assembler routine)                                    *
*                                                                           *
* Author      = Russ Hougland                                               *
*                                                                           *
* Function    = To draw a line (horizontal or vertical) directly into the   *
*               video memory.  It will not draw a diagonal line.  This      *
*               routine allows the user to specify the beginning and ending *
*               characters of the line (in case it intercepts a box).       *
*                                                                           *
* Usage       = integer*2 r1, c1, r2, c2, begchr, endchr, type, attr        *
*               call line (r1, c1, r2, c2, begchr, endchr, type, attr)      *
*                                                                           *
* Parameters =                                                              *
*      r1     - starting row on the screen (1-25)                          *
*      c1     - starting column on the screen (1-80)                       *
*      r2     - ending row on the screen (1-25)                            *
*      c2     - ending column on the screen (1-80)                         *
*     begchr  - the beginning character of the line (the decimal ascii value) *
*     endchr  - the ending character of the line (the decimal ascii value)  *
*     type    - the type of line to draw                                    *
*               1 - single line                                             *
*               2 - double line                                             *
*     attr    - attribute in which to display the line (see wt)             *
*                                                                           *
* Returns     = None.                                                       *
*                                                                           *
* Calls       = None.                                                       *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*************************************************************************
* Name        = pi (prompt for integer)                                 *
*                                                                       *
* Author      = Kevin Stewart                                           *
*                                                                       *
* Function    = To write out a prompt and accept an integer response.  After *
*               accepting the response, the prompt and reply are erased.  *
*               The routine places one space between the end of the prompt *
*               and the start of the reply.                             *
*                                                                       *
* Usage       = integer*2 row, col, rlen, code, patt, ratt              *
*               integer*4 reply                                         *
*               character prompt*(*)                                    *
*               call pi (row, col, prompt, reply, rlen, code, patt, ratt) *
*                                                                       *
* Parameters =                                                          *
*     row     - the row to display the prompt on                        *
*     col     - the column to begin displaying the prompt on            *
*               0 (center the prompt and reply)                         *
*     prompt  - the text to be displayed as a prompt                    *
*     reply   - the integer reply of the user                           *
*     rlen    - the maximum length of the user's reply                  *
*     code    - code can give special instructions for the reply entry. *
*               more than one can be passed by summing the options desired. *
*               4 (user must enter a carriage return to enter a value)  *
*               8 (the number will not be written upon entry and exit from ri) *
*               16 (the prompt will not be erased upon exit from pi)    *
*     patt    - the attribute to display the prompt in (see wt)         *
*     ratt    - the attribute to display the reply in (see wt)          *
*                                                                       *
* Returns     = reply                                                   *
*                                                                       *
* Calls       = fsc.lib:  chkbit ri  window  wt                         *
*               fortran:  len                                           *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name       = pt (prompt for text)                                     *
*                                                                       *
* Author     = Kevin Stewart                                            *
*                                                                       *
* Function   = To write out a prompt and accept a text response.  After *
*              accepting the response, the prompt and reply are erased. *
*              The routine places one space between the end of the prompt *
*              and the start of the reply.                              *
*                                                                       *
* Usage      = integer*2 row, col, code, patt, ratt                     *
*              character prompt*(*), reply*(*)                          *
*              call pt (row, col, prompt, reply, code, patt, ratt)      *
*                                                                       *
* Parameters =                                                          *
*     row    - the row to display the prompt on                         *
*     col    - the column to begin displaying the prompt on             *
*              0 (center the prompt and reply)                          *
*     prompt - the text to be displayed as a prompt                     *
*     reply  - the text reply of the user                               *
*     code   - code can give special instructions for the reply entry.  *
*              more than one can be passed by summing the options desired. *
*              1 (the string is made upper case)                        *
*              2 (the string is packed -- no blanks)                    *
*              4 (user must enter a carriage return to enter a value)   *
*              8 (the text will not be written upon entry and exit from rt) *
*              16 (the prompt will not be erased upon exit from pi)     *
*     patt   - the attribute to display the prompt in (see wt)          *
*     ratt   - the attribute to display the reply in (see wt)           *
*                                                                       *
* Returns    = reply                                                    *
*                                                                       *
* Calls      = fsc.lib:  chkbit  rt  window  wt                         *
*              fortran:  len                                            *
*                                                                       *
* Commons    = None.                                                    *
*************************************************************************


*************************************************************************
* Name       = rconv (convert character string to real)                 *
*                                                                       *
* Author     = Kevin Stewart                                            *
*                                                                       *
* Function   = To convert a character string of numbers to its real     *
*              equivalent.  The largest number that can be converted can *
*              have up to 10 digits and 10 decimal places.              *
*                                                                       *
* Usage      = character str*(*)                                        *
*              real*8    num, rconv                                     *
*              logical*2 err                                            *
*              num = rconv (str, err)                                   *
*                                                                       *
* Parameters =                                                          *
*       str - the character string containing the number to convert     *
*       err - true if an error occurrs                                  *
*                                                                       *
* Returns    = rconv and err (if err, then rconv equals 0)              *
*                                                                       *
* Calls      = fsc.lib:  iconv  compac                                  *
*                                                                       *
* Commons    = None.                                                    *
*************************************************************************
```

```
*****************************************************************************
* Name        = rd (read date)                                              *
*                                                                           *
* Author      = Russ Hougland                                               *
*                                                                           *
* Function    = Reads a user's reply (date field) at a specified location   *
*               (row,col).  Before the reply is read, the field is written  *
*               in the attribute specified (attin); upon termination, the   *
*               reply is rewritten in the attribute specified (attout).     *
*               The date is always returned in internal format              *
*               ('YYYY/0MM/DD').  This routine returns either a blank date  *
*               ('0000/000/00') or a valid date, but never an invalid date. *
*                                                                           *
* Usage       = integer*2 row, col, code, attin, attout                     *
*               character fmt*01, date*11                                   *
*               call rd (row, col, date, fmt, code, attin, attout)          *
*                                                                           *
* Parameters =                                                              *
*     row     - the row on which to read the date                           *
*     col     - the column on which to read the date                        *
*     date    - the date to be read                                         *
*     fmt     - the format of date for display and entry purposes (any case)*
*               'M' (metric DD/MMM/YYYY)                                    *
*               'E' (english MMM/DD/YYYY)                                   *
*     code    - code can give special instructions for the date entry.      *
*               more than one can be passed by summing the options desired. *
*               4 (user must enter a carriage return to enter a value)      *
*               8 (the date will not be written upon entry and exit from rd)*
*     attin   - the attribute to display the date in upon entry (see wt)    *
*     attout  - the attribute to display the date in upon exit (see wt)     *
*                                                                           *
* Returns     = code - indicates how rd was terminated (e.g  carriage return,*
*               up arrow, etc.)                                             *
*               date - the date entered by the user                         *
*                                                                           *
* Calls       = fsc.lib:  chkbit  window  ri  wi  rt  wt  wd                *
*               fortran:  ichar   mod                                       *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*************************************************************************
* Name       = rf (read fixed real)                                     *
*                                                                       *
* Author     = Bobby Adams                                              *
*                                                                       *
* Function   = Reads a user's reply (real field) at a specified location *
*              (row,col).  Before the reply is read, the field can be   *
*              written in a specified attribute (attin); upon termination,*
*              the reply can be rewritten in a specified attribute (attout).*
*              Note the real number read is double precision.  Further, the*
*              total length of the number is len1 + len2 + 1 (decimal point).*
*                                                                       *
* Usage      = integer*2 row, col, len1, len2, code, attin, attout      *
*              real*8    num                                          *
*              call rf (row, col, num, len1, len2, code, attin, attout) *
*                                                                       *
* Parameters =                                                          *
*     row    - the row on which to read the number                      *
*     col    - the column on which to read the number                   *
*     num    - the number to be read                                    *
*     len1   - the maximum length of the left side of the number        *
*     len2   - the maximum length of the right side of the number       *
*     code   - code can give special instructions for the number entry. *
*              more than one can be passed by summing the options desired.*
*              4 (user must enter a carriage return to enter a value)   *
*              8 (the number will not be written upon entry and exit from rf)*
*     attin  - the attribute to display the number in upon entry (see wt)*
*     attout - the attribute to display the number in upon exit (see wt)*
*                                                                       *
* Returns    = code - indicates how rf was terminated (e.g. carriage return,*
*              up arrow, etc.)                                          *
*              num  - the number entered by the user                    *
*                                                                       *
* Calls      = fsc.lib:  chkbit  window  cursor  csrpos  inkey  wt  wf  *
*                                                                       *
* Commons    = None.                                                    *
*************************************************************************
```

```
*************************************************************************
* Name        = ri (read integer)                                       *
*                                                                       *
* Author      = Bobby Adams                                             *
*                                                                       *
* Function    = Reads a user's reply (integer field) at a specified location *
*               (row,col).  Before the reply is read, the field can be  *
*               written in a specified attribute (attin); upon termination, *
*               the reply can be rewritten in a specified attribute (attout). *
*                                                                       *
* Usage       = integer*2 row, col, len, code, attin, attout            *
*               integer*4 num                                           *
*               call ri(row, col, num, len, code, attin, attout)        *
*                                                                       *
* Parameters =                                                          *
*      row    - the row on which to read the number                     *
*      col    - the column on which to read the number                  *
*      num    - the number to be read                                   *
*      len    - the maximum length of the number                        *
*     code    - code can give special instructions for the number entry. *
*               more than one can be passed by summing the options desired. *
*               4 (user must enter a carriage return to enter a value)  *
*               8 (the number will not be written upon entry and exit from ri) *
*     attin   - the attribute to display the number in upon entry (see wt) *
*     attout  - the attribute to display the number in upon exit (see wt) *
*                                                                       *
* Returns     = code - indicates how ri was terminated (e.g. carriage return, *
*               up arrow, etc.)                                         *
*               num  - the number entered by the user                   *
*                                                                       *
* Calls       = fsc.lib:  chkbit  cursor  window  csrpos  inkey  wi  wt *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = rpterr (report error)                                   *
*                                                                       *
* Author      = Kevin Stewart                                           *
*                                                                       *
* Function    = To append an error message to the report file (error.dat). *
*                                                                       *
* Usage       = character sn*(*), msg*(*)                               *
*               call rpterr (sn, msg)                                   *
*                                                                       *
* Parameters =                                                          *
*        sn   - subroutine name where error occurred                    *
*       msg   - error message to written to the error file              *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  sysdat  cd                                    *
*               fortran:  gettim  inquire  open  close                  *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = rr (read real)                                          *
*                                                                       *
* Author      = Bobby Adams                                             *
*                                                                       *
* Function    = Reads a user's reply (real field) at a specified location *
*               (row,col).  Before the reply is read, the field can be  *
*               written in a specified attribute (attin); upon termination, *
*               the reply can be rewritten in a specified attribute (attout). *
*                                                                       *
* Usage       = integer*2 row, col, len, code, attin, attout            *
*               real*4    num                                         *
*               call rr(row, col, num, len, code, attin, attout)        *
*                                                                       *
* Parameters =                                                          *
*      row    - the row on which to read the number                     *
*      col    - the column on which to read the number                  *
*      num    - the number to be read                                   *
*      len    - the maximum length of the number                        *
*     code    - code can give special instructions for the number entry. *
*               more than one can be passed by summing the options desired. *
*               4 (user must enter a carriage return to enter a value)  *
*               8 (the number will not be written upon entry and exit from rr) *
*     attin   - the attribute to display the number in upon entry (see wt) *
*     attout  - the attribute to display the number in upon exit (see wt) *
*                                                                       *
* Returns     = code - indicates how rr was terminated (e.g. carriage return, *
*               up arrow, etc.)                                         *
*               num  - the number entered by the user                   *
*                                                                       *
* Calls       = fsc.lib:  chkbit  window  cursor  csrpos  inkey  wt  wr *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = rsscrn (assembler routine)                              *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Restores the screen display previously saved by svscrn.  *
*                                                                       *
* Usage       = call rsscrn                                             *
*                                                                       *
* Parameters = None                                                     *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name       = rt (read text)                                           *
*                                                                       *
* Author     = Russ Hougland                                            *
*                                                                       *
* Function   = Reads a user's reply (text field) at a specified location *
*              (row,col).  Before the reply is read, the field can be   *
*              written in a specified attribute (attin); upon termination,*
*              the reply can be rewritten in a specified attribute (attout).*
*                                                                       *
* Usage      = integer*2 row, col, code, attin, attout                  *
*              character str*(*)                                        *
*              call rt (row, col, str, code, attin, attout)             *
*                                                                       *
* Parameters =                                                          *
*      row   - the row on which to read the string                      *
*      col   - the column on which to read the string                   *
*      str   - the string to be read                                    *
*     code   - code can give special instructions for the text entry.   *
*              more than one can be passed by summing the options desired.*
*              1 (the string is made upper case)                        *
*              2 (the string is packed -- no blanks)                    *
*              4 (user must enter a carriage return to enter a value)   *
*              8 (the number will not be written upon entry and exit from rt)*
*     attin  - the attribute to display the string in upon entry (see wt)*
*     attout - the attribute to display the string in upon exit (see wt)*
*                                                                       *
* Returns    = code - indicates how rt was terminated (e.g. carriage return,*
*              up arrow, etc.)                                          *
*              str  - the string entered by the user                    *
*                                                                       *
* Calls      = fsc.lib:  chkbit  cursor  csrpos  compac  inkey  wt      *
*              fortran:  ichar                                          *
*                                                                       *
* Commons    = None.                                                    *
*************************************************************************
```

```
*************************************************************************
* Name        = rv (assembler routine)                                  *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = To convert an attribute to reverse video.  It reverses the *
*               foreground and background RGB bits while leaving the intensity *
*               and blinking bits unchanged.                            *
*                                                                       *
* Usage       = integer*2 newatt, oldatt, rv                            *
*               newatt = rv (oldatt)                                    *
*                                                                       *
* Parameters =                                                          *
*     oldatt - the old attribute that needs reversing                   *
*                                                                       *
* Returns     = rv (the new attribute after reversing).                 *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = scroll (assembler routine)                              *
*                                                                       *
* Author      = Dan Weidenfeld                                          *
*                                                                       *
* Function    = To scroll a window on the screen.                       *
*                                                                       *
* Usage       = integer*2 r1, c1, r2, c2, action, att                   *
*               call scroll (r1, c1, r2, c2, action, att)               *
*                                                                       *
* Parameters =                                                          *
*     r1      - row of the upper left hand corner of the window to be   *
*               scrolled - 1 (0-24)                                     *
*     c1      - column of the upper left hand corner of the window to be *
*               scrolled - 1 (0-79)                                     *
*     r2      - row of the lower right hand corner of the window to be  *
*               scrolled - 1 (0-24)                                     *
*     c2      - column of the lower right hand corner of the window to be *
*               scrolled - 1 (0-79)                                     *
*     action - the action to perform                                    *
*               +n (scroll up n lines)                                  *
*               -n (scroll down n lines)                                *
*                0 (scroll the entire window)                           *
*     att     - the attribute of blank lines added after scrolling (see wt) *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = setatt                                                  *
*                                                                       *
* Author      = Bobby Adams                                             *
*                                                                       *
* Function    = To initialize the values of the attribute variables used by *
*               the screen read and write routines.  This routine must be *
*               called once at the beginning of any program that uses the *
*               attribute variables stored in attrib.inc.  The color values of *
*               the variables can be changed by changing the file, color.dat. *
*                                                                       *
* Usage       = call setatt                                             *
*                                                                       *
* Parameters = None.                                                    *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       - fsc.lib:  clrmod                                        *
*                                                                       *
* Commons     = attrib (attrib.inc) - contains the values to initialize *
*************************************************************************

*************************************************************************
* Name        = slen (integer function)                                 *
*                                                                       *
* Author      = Kevin Stewart                                           *
*                                                                       *
* Function    = Find the length of a character string within a character *
*               variable (i.e. the position of the last non-blank character). *
*                                                                       *
* Usage       = integer*2 i, slen                                       *
*               character str*(*)                                       *
*               i = slen (str)                                          *
*                                                                       *
* Parameters =                                                          *
*        str - character variable containing the string being passed    *
*                                                                       *
* Returns     = slen - the length of the character string (0 if all blanks) *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************


*************************************************************************
* Name        = svscrn (assembler routine)                              *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Saves the screen display in a buffer area for later     *
*               restoration by rsscrn.                                  *
*                                                                       *
* Usage       = call svscrn                                             *
*                                                                       *
* Parameters = None                                                     *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*****************************************************************************
* Name        = sysdat (system date)                                        *
*                                                                           *
* Author      = Kevin Stewart                                               *
*                                                                           *
* Function    = To return the system date in the internal storage format.   *
*                                                                           *
* Usage       = character*11 dat                                            *
*               call sysdat (dat)                                           *
*                                                                           *
* Parameters =                                                              *
*         dat - the system date in internal format ('YYYY/MMM/DD')          *
*                                                                           *
* Returns     = the system date (dat).                                      *
*                                                                           *
* Calls       = fortran:  getdat                                            *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************


*****************************************************************************
* Name        = today (assembler routine)                                   *
*                                                                           *
* Author      = Dan Weidenfeld                                              *
*                                                                           *
* Function    = To get the system date.                                     *
*                                                                           *
* Usage       = integer*2 year, month, day                                  *
*               call today (year, month, day)                               *
*                                                                           *
* Parameters =                                                              *
*       year  - the year of the system date (i.e. 1986)                     *
*       month - the month of the system date (i.e. 12)                      *
*       day   - the day of the system date (i.e. 25)                        *
*                                                                           *
* Returns     = the system date (year,month,day).                           *
*                                                                           *
* Calls       = None.                                                       *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************


*****************************************************************************
* Name        = upper                                                       *
*                                                                           *
* Author      = Russ Hougland                                               *
*                                                                           *
* Function    = Converts to upper case all letters in the parameter.        *
*                                                                           *
* Usage       = character*(*) str                                           *
*               call upper (str)                                            *
*                                                                           *
* Parameters =                                                              *
*         str - the character string to be converted to upper case          *
*                                                                           *
* Returns     = str (all letters in upper case)                             *
*                                                                           *
* Calls       = fortran:  ichar  char                                       *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*************************************************************************
* Name        = wd (write date)                                         *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Write a date at a specified location (row,col) in the form *
*               specified.  The date should be passed in the internal format *
*               ('YYYY/MMM/DD').                                        *
*                                                                       *
* Usage       = integer*2 row, col, attr                                *
*               character fmt*01, date*11                               *
*               call wd (row, col, date, fmt, attr)                     *
*                                                                       *
* Parameters =                                                          *
*      row    - the row on which to write the date                      *
*      col    - the column on which to write the date                   *
*               0 (to center the date on the line)                      *
*      date   - the date to be written                                  *
*      fmt    - the format of date (can be lower or upper case)         *
*               'M' (metric DD/MMM/YYYY)                                *
*               'E' (english MMM/DD/YYYY)                               *
*      attr   - the attribute to display the date in (see wt)           *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  cd  wt                                        *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = wf (write fixed real)                                   *
*                                                                       *
* Author      = Bobby Adams                                             *
*                                                                       *
* Function    = Write a real at a specified location (row,col) using the *
*               desired attribute.  Note the number written is double   *
*               precision.  Further, the total length of the number is  *
*               len1 + len2 + 1 (decimal point).                        *
*                                                                       *
* Usage       = integer*2 row, col, attr, len1, len2                    *
*               real*8    num                                           *
*               call wf (row, col, num, len1, len2, attr)               *
*                                                                       *
* Parameters =                                                          *
*      row    - the row on which to write the number                    *
*      col    - the column on which to write the number                 *
*               0 (to center the number on the line)                    *
*      num    - the number to be written                                *
*      len1   - the length of the left side of the number in            *
*      len2   - the length of the right side of the number in           *
*      attr   - the attribute to display the date in (see wt)           *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  compac  wt                                    *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = wi (write integer)                                      *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Write an integer at a specified location (row,col) using the  *
*               desired attribute.                                      *
*                                                                       *
* Usage       = integer*2 row, col, attr, len                           *
*               integer*4 num                                           *
*               call wi (row, col, num, len, attr)                      *
*                                                                       *
* Parameters =                                                          *
*      row    - the row on which to write the number                    *
*      col    - the column on which to write the number                 *
*               0 (to center the number on the line)                    *
*      num    - the number to be written                                *
*      len    - the length of the field to display the number in        *
*      attr   - the attribute to display the number in (see wt)         *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  wi                                            *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = window (assembler routine)                              *
*                                                                       *
* Author      = Russ Hougland                                           *
*                                                                       *
* Function    = Set the video attribute for a window directly into video  *
*               memory.                                                 *
*                                                                       *
* Usage       = integer*2 r1, c1, r2, c2, attr                          *
*               call window (r1, c1, r2, c2, attr)                      *
*                                                                       *
* Parameters =                                                          *
*       r1    - row of the upper left hand corner                       *
*       c1    - column of the upper left hand corner                    *
*       r2    - row of the lower right hand corner (x2 must be > x1)    *
*       c2    - column of the lower right hand corner (y2 must be > y1)  *
*       attr  - the attribute of the window (see wt)                    *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = None.                                                   *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = wr (write real)                                         *
*                                                                       *
* Author      = Bobby Adams                                             *
*                                                                       *
* Function    = Write a real at a specified location (row,col) using the*
*               desired attribute.                                      *
*                                                                       *
* Usage       = integer*2 row, col, attr, len                           *
*               real*4    num                                           *
*               call wr (row, col, num, len, attr)                      *
*                                                                       *
* Parameters =                                                          *
*      row    - the row on which to write the number                    *
*      col    - the column on which to write the number                 *
*               0 (to center the number on the line)                    *
*      num    - the number to be written                                *
*      len    - the length of the field to display the number in        *
*      attr   - the attribute to display the date in (see wt)           *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  wt                                            *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = wsd (write screen date)                                 *
*                                                                       *
* Author      = Kevin Stewart                                           *
*                                                                       *
* Function    = To write a prompt and a variable to the screen.         *
*                                                                       *
* Usage       = integer*2 row, col, patt, vatt                          *
*               character prompt*(*), dat*11, fmt*01                    *
*               call wsd (row, col, prompt, dat, fmt, patt, vatt)       *
*                                                                       *
* Parameters =                                                          *
*     row     - the row position of the date                            *
*     col     - the column position of the date                         *
*               0 (center the prompt and the date in the row)           *
*     prompt  - the prompt (displayed at row,col-plen-1)                *
*     dat     - the date variable to display                            *
*     fmt     - the format to display the date in (english or metric)   *
*     patt    - the attribute to use when displaying the prompt         *
*     vatt    - the attribute to use when displaying the date           *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  wd  wt                                        *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = wsf (write screen fixed real)                           *
*                                                                       *
* Author      = Kevin Stewart                                           *
*                                                                       *
* Function    = To write a prompt and a variable to the screen.         *
*                                                                       *
* Usage       = integer*2 row, col, len1, len2, patt, vatt              *
*               character prompt*(*)                                    *
*               real*8    var                                           *
*               call wsf (row, col, prompt, var, len1, len2, patt, vatt)*
*                                                                       *
* Parameters =                                                          *
*     row     - the row position of the variable                        *
*     col     - the column position of the variable                     *
*               0 (center the prompt and the variable in the row)       *
*     prompt  - the prompt (displayed at row,col-plen-1)                *
*     var     - the variable to display                                 *
*     len1    - the maximum length of the left side of the number       *
*     len2    - the maximum length of the right side of the number      *
*     patt    - the attribute to use when displaying the prompt         *
*     vatt    - the attribute to use when displaying the variable       *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  wf  wt                                        *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*************************************************************************
* Name        = wsi (write screen integer)                              *
*                                                                       *
* Author      = Kevin Stewart                                           *
*                                                                       *
* Function    = To write a prompt and a variable to the screen.         *
*                                                                       *
* Usage       = integer*2 row, col, vlen, patt, vatt                    *
*               character prompt*(*)                                    *
*               integer*4 var                                           *
*               call wsi (row, col, pro  , var, vlen, patt, vatt)       *
*                                                                       *
* Parameters =                                                          *
*     row     - the row position of the variable                        *
*     col     - the column position of the variable                     *
*               0 (center the prompt and the variable in the row)       *
*     prompt  - the prompt (displayed at row,col-plen-1)                *
*     var     - the variable to display                                 *
*     vlen    - the length of the variable                              *
*     patt    - the attribute to use when displaying the prompt         *
*     vatt    - the attribute to use when displaying the variable       *
*                                                                       *
* Returns     = None.                                                   *
*                                                                       *
* Calls       = fsc.lib:  wi  wt                                        *
*                                                                       *
* Commons     = None.                                                   *
*************************************************************************
```

```
*****************************************************************************
* Name        = wsr (write screen real)                                     *
*                                                                           *
* Author      = Kevin Stewart                                               *
*                                                                           *
* Function    = To write a prompt and a variable to the screen.             *
*                                                                           *
* Usage       = integer*2 row, col, vlen, patt, vatt                        *
*               character prompt*(*)                                        *
*               real*4    var                                               *
*               call wsr (row, col, prompt, var, vlen, patt, vatt)          *
*                                                                           *
* Parameters =                                                              *
*     row     - the row position of the variable                            *
*     col     - the column position of the variable                         *
*               0 (center the prompt and the variable in the row)           *
*     prompt  - the prompt (displayed at row,col-plen-1)                    *
*     var     - the variable to display                                     *
*     vlen    - the length of the variable                                  *
*     patt    - the attribute to use when displaying the prompt             *
*     vatt    - the attribute to use when displaying the variable           *
*                                                                           *
* Returns     = None.                                                       *
*                                                                           *
* Calls       = fsc.lib:  wr  wt                                            *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*****************************************************************************
* Name        = wst (write screen text)                                     *
*                                                                           *
* Author      = Kevin Stewart                                               *
*                                                                           *
* Function    = To write a prompt and a variable to the screen.             *
*                                                                           *
* Usage       = integer*2 row, col, patt, vatt                              *
*               character prompt*(*), var*(*)                               *
*               call wst (row, col, prompt, var, patt, vatt)                *
*                                                                           *
* Parameters =                                                              *
*     row     - the row position of the variable                            *
*     col     - the column position of the variable                         *
*               0 (center the prompt and the variable in the row)           *
*     prompt  - the prompt (displayed at row,col-plen-1)                    *
*     var     - the variable to display                                     *
*     patt    - the attribute to use when displaying the prompt             *
*     vatt    - the attribute to use when displaying the variable           *
*                                                                           *
* Returns     = None.                                                       *
*                                                                           *
* Calls       = fsc.lib:  wt                                                *
*                                                                           *
* Commons     = None.                                                       *
*****************************************************************************
```

```
*************************************************************************
* Name       = wt (write text -- assembler routine)                     *
*                                                                       *
* Author     = Russ Hougland                                            *
*                                                                       *
* Function   = Write a string at a specified location (row,col) using the *
*              desired attribute.                                       *
*                                                                       *
* Usage      = integer*2 row, col, attr, len                            *
*              character str*(*)                                        *
*              call wt (row, col, str, len, attr)                       *
*                                                                       *
* Parameters =                                                          *
*      row   - the row on which to write the string                     *
*      col   - the column on which to write the string                  *
*              0  (center the field including trailing blanks)          *
*      str   - the string to be written                                 *
*      len   - the length of the field to display the string in         *
*      attr  - the attribute to display the string in                   *
*              to determine the number for a color use the following:   *
*                 foreground color:  blue    1                         *
*                                    green   2                          *
*                                    red     4                          *
*                 intensity       :          8                         *
*                 background color:  blue   16                         *
*                                    green  32                          *
*                                    red    64                          *
*                 blinking        :       128                         *
*                                                                       *
*              to get the desired color simply sum the numbers of the desired *
*              attributes, for example: to get a blinking white text on a *
*              black background attr = 1 + 2 + 4 (white foreground)     *
*                                    + 0         (black background)     *
*                                    + 128       (blinking)             *
*                                                                       *
* Returns    = None.                                                    *
*                                                                       *
* Calls      = None.                                                    *
*                                                                       *
* Commons    = None.                                                    *
*************************************************************************
```

BTRIEVE CALLS

```
FUNCTION   :  INTERFACE WITH THE BTRIEVE TO ACCESS A SPECIFIED FILE

USAGE      :  INTEGER*2 STS,FILBLK(64),REC(N),LREC,LKEY,OP,KEYNUM
              CHARACTER KEY*N
              CALL BTREE(FILBLK,OP,KEYNUM,KEY,LKEY,REC,LREC,STS)

PARAMETERS :  FILBLK - STORES THE FILE NAME AND PATH, CURRENT RECORD
                       POINTER

              OP     - THE TYPE OF OPERATION TO BE EXECUTED
                       (i.e. OPEN,UPDATE, ect.)

              KEYNUM - THE KEY PATH USED IN ALL FILE OPERATIONS
                                  ** NOTE **
                       THE KEYNUM USED MUST BE A VALID KEY SPECIFED
                       DURING THE FILES INITIAL OPERATION.

              KEY    - CONTAINS THE KEY FIELD OF THE RECORD

              LKEY   - LENGTH OF KEY FIELD IN BYTES

              REC    - INTEGER*2 ARRAY CONTAINING RECORD DATA

              LREC   - LENGTH OF RECORD IN BYTES

              STS    - CONTAINS THE STATUS OF THE OPERATION PERFORMED
                       RETURN ZERO IF NORMAL OPERATION

NOTE       :  WHEN CREATING AND OPENING FILES, THE FILE NAME IS IN THE
              'KEY' FIELD AND THE FILE NAME LENGTH IS IN THE 'LKEY'
              FILED, example:

              CALL BTREE (FILBLK,0,0,'FILE.XDB',8,REC,LREC,STS)
```

BTRIEVE OPERATIONS (for detail information see Btrieve menu Chap. 9)

```
(0)       - OPEN
PURPOSE   - TO OPEN A BTRIEVE FILE

(1)       - CLOSE
PURPOSE   - TO CLOSE A BTRIEVE FILE

(2)       - INSERT
PURPOSE   - TO INSERT A RECORD IN A FILE

(3)       - UPDATE
PURPOSE   - TO UPDATE AN EXISTING RECORD IN A BTRIEVE FILE

(4)       - DELETE
PURPOSE   - TO DELETE AN EXISTING RECORD IN A BTRIEVE FILE

(5)       - GET EQUAL
PURPOSE   - TO RETRIEVE A RECORD FROM A BTRIEVE FILE CORRESPONDING
            TO A SPECIFIC KEY VALUE

(6)       - GET NEXT
PURPOSE   - TO RETRIEVE A RECORD FROM A BTRIEVE FILE WHICH FOLLOWS
            THE "CURRENT RECORD"
```

(7) - GET PREVIOUS
PURPOSE - TO RETRIEVE A RECORD FROM A BTRIEVE FILE WHICH PRECEDES THE "CURRENT RECORD"

(8) - GET GREATER
PURPOSE - TO RETRIEVE A RECORD FROM A BTRIEVE FILE CORRESPONDING TO THE KEY VALUE WHICH IS GREATER THAN A SPECIFIC KEY VALUE

(9) - GET GREATER OR EQUAL
PURPOSE - TO RETRIEVE A RECORD FROM A BTRIEVE FILE CORRESPONDING TO THE KEY VALUE WHICH IS GREATER OR EQUAL TO A SPECIFIC KEY VALUE

(10) - GET LESS THAN
PURPOSE - TO RETRIEVE A RECORD FROM A BTRIEVE FILE CORRESPONDING TO THE KEY VALUE WHICH IS LESS THAN A SPECIFIC KEY VALUE

(11) - GET LESS THAN OR EQUAL
PURPOSE - TO RETRIEVE A RECORD FROM A BTRIEVE FILE CORRESPONDING TO THE KEY VALUE WHICH IS LESS THAN OR EQUAL TO A SPECIFIC KEY VALUE

(12) - GET LOWEST
PURPOSE - TO RETRIEVE A RECORD FROM A BTRIEVE FILE CORRESPONDING TO THE LOWEST KEY VALUE FOR A SPECIFIED ACCESS PATH.

(13) - GET HIGHEST
PURPOSE - TO RETRIEVE A RECORD FROM A BTRIEVE FILE CORRESPONDING TO THE HIGHEST KEY VALUE FOR A SPECIFIED ACCESS PATH.

(14) - CREATE
PURPOSE - TO CREAT A BTRIEVE FILE WITH THE SPECIFIED SET OF CHARACTERISTICS

(15) - STAT
PURPOSE - TO RETRIEVE A SPECIFIED FILE'S CHARACTERISTICS

(16) - EXTEND
PURPOSE - TO EXTEND A BTRIEVE FILE TO A SECOND LOGICAL DISK DRIVE

(17) - SET DIRECTORY
PURPOSE - TO SET THE CURRENT DIRECTORY TO A SPECIFIED VALUE

(18) - GET DIRECTORY
PURPOSE - TO RETRIEVE THE "CURRENT" DIRECTORY

(19) - BEGIN TRANSACTION
PURPOSE - TO MARK THE BEGINNING OF A SET OF LOGICALLY RELATED BTRIEVE OPERATIONS.

(20) - END TRANSACTION
PURPOSE - TO COMPLETE A TRANSACTION AND COMMIT THE OPERATIONS PERFORMED SINCE THE TRANSACTION BEGAN

(21) - ABORT TRANSACTION
PURPOSE - TO REMOVE ALL OPERATIONS PERFORMED SINCE THE BEGINNING OF AN ACTIVE TRANSACTION AND TO TERMINATE THE TRANSACTION.

(22) - GET POSITION
PURPOSE - TO RETURN THE PHYSICAL POSITION OF THE RECORD IN THE BTRIEVE FILE THAT HAS BEEN ESTABLISHED AS THE "CURRENT RECORD"

(23) - GET DIRECT
PURPOSE - TO RETRIEVE THE DATA RECORD POSITIONED AT A SPECIFED ADDRESS IN THE BTRIEVE FILE.

(24) - STEP DIRECT
PURPOSE - TO RETRIEVE THE DATA RECORD IN THE LOCATION PHYSICALLY FOLLOWING THE CURRENT RECORD IN THE BTRIEVE FILE.

(25) - STOP
PURPOSE - TO TERMINATE THE RECORD MANAGER AND REMOVE IT FROM MEMORY.

BTRIEVE ERROR CODES

(For detailed information see Btrieve menu Appendix B ERROR CONDITIONS)

A utility program (STS.EXE) is available to prompt Btrieve Error Condition and Fortran Runtime Error.

01	INVALID OPERATION
02	I/O ERROR
03	NO OPEN
04	KEY NOT FOUND
05	DUPLICATES ERROR
06	INVALID KEY NUMBER
07	DIFFERENT KEY NUMBER
08	INVALID POSITIONING
09	END OF FILE
10	MODIFIABLE ERROR
11	INVALID FILE NAME
12	FILE NOT FOUND
13	EXTENSION ERROR
14	PRE-OPEN ERROR
15	PRE-IMAGE ERROR
16	EXPANSION ERROR
17	CLOSE ERROR
18	DISK FULL
19	UNRECOVERABLE ERROR
20	RECORD MANAGER INACTIVE
21	KEY BUFFER ERROR
22	RECORD BUFFER (DATA BUFFER NOT LONG ENOUGH)
23	POSITION BLOCK (MUST BE 128 BYTES)
24	PAGE SIZE (MUST BE MULTIPLE OF 512)
25	CREATE I/O ERROR
26	NUMBER OF KEYS (PAGE SIZE AND NUM OF KEYS DO NOT MATCH)
27	KEY POSITION
28	RECORD LENGTH (NO GREATER THAN PAGE SIZE - 6)
29	KEY LENGTH (1-255)
30	BTRIEVE FILE NAME (INVALID BTRIEVE FILE)
31	EXTEND ERROR (ALREADY EXTENDED)
32	EXTEND I/O ERROR
33	EXTEND DRIVE ERROR
34	EXTEND NAME
35	DIRECTORY ERROR
36	TRANSACTON ERROR (/T OPTION NOT SPECIFIED WHEN REC-MAN LOADED)
37	BEGIN TRANSACTION (TRANSACTION ALREADY ACTIVE)
38	TRANSACTION CONTROL FILE
39	END/ABORT ERROR
40	TRANSACTION MAX FILES (UP TO 8 FILES MAY BE UPDATED DURING TRANS)
41	TRANSACTION OPEN/CLOSE
42	INCOMPLETE ACCELERATED ACCESS
43	INVALID DATA RECORD ADDRESS
44	NULL KEY PATH
45	INCONSISTENT KEY FLAGS
46	ACCESS DENIED
47	MAXIMUM OPEN FILES
48	INVALID ALTERNAMTE SEQUENCE DEFINITION
49	KEY TYPE ERROR
50	OWNER ALREADY SET
51	INVALID OWNER
52	ERROR WRITING CACHE
53	INVALID INTERFACE
54	VARIABLE PAGE UNREADABLE
80	CONFLICT
81	LOCK FULL

```
82      LOST POSITION
83      READ OUTSIDE TRANSACTION
84      RECORD IN USE
85      FILE IN USE
86      FILE FULL
87      HANDLE FULL
88      MODE ERROR
89      NAME ERROR
90      DEVICE FULL
91      SERVER ERROR
92      TRANSACTION FULL
99      DEMO ERROR
```

5 STANDARD COMMON BLOCKS

Table 3 contains an alphabetical list of the standard common blocks and a short description.

Table 3

Standard Common Blocks

```
*********************************************************

 FILE NAME: AMSCOD.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: AMSCOD.INC
 NUMBER OF KEYS:  2
 RECORD SIZE : 52
 PAGE SIZE   : 512
 RECORD NAME : AMSREC

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     AMSKEY      1         2     N   N    S
  1     AMSCOD      3         10    N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 AMSKEY          S   2          1       AMS ID
 AMSCOD          S   10         3       AMS CODE
 AMSDSC          S   40         13      AMS DESCRIPTION

*********************************************************

 FILE NAME: AMSDSC.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: AMSDSC.INC
 NUMBER OF KEYS:  2
 RECORD SIZE : 50
 PAGE SIZE   : 512
 RECORD NAME : AMSRC

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     AMSID       1        10    N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 AMSID           S   10         1       AMS CODE
 AMSDSC          S   40         11      AMS DESCRIPTION

*********************************************************
```

```
*********************************************************

 FILE NAME: AMSF4C.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: AMSF4C.INC
 NUMBER OF KEYS:  2
 RECORD SIZE : 18
 PAGE SIZE   : 512
 RECORD NAME : F4CREC

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     F4CCOD      1         7     N   N    S
  1     AMSEQV      9         10    Y   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 F4CCOD          S   7          1       F4C CODE
 AMSEQV          S   10         9       AMS NUMBER

*********************************************************

 FILE NAME: APRCOD.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: APRCOD.INC
 NUMBER OF KEYS:  2
 RECORD SIZE : 52
 PAGE SIZE   : 512
 RECORD NAME : APRREC

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     APRKEY      1         2     N   Y    S
  1     APRCOD      3         10    N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 APRKEY          S   2          1       APPROPRIATION ID
 APRCOD          S   10         3       APP. CODE
 APRDSC          S   40         13      APP. DESCRIPTION

*********************************************************
```

```
**********************************************************

 FILE NAME: AREA_TAB.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 34
 PAGE SIZE   : 512
 RECORD NAME : ARDAT

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     CODE         1        2     N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 CODE            S   2           1       AREA ID
 DEF             S   30          3       AREA DESCRIPTION

**********************************************************

 FILE NAME: BT-PASS.DAT
 FILE TYPE: TEXT
 NUMBER OF LINES: 1

 **** FILE INFORMATION ****
  LINE     FORMAT     FIELDS
 *****   **********  **********
   1         A4         TTGG

 ***** FIELD INFORMATION *****
 FIELD          TYPE LENGTH  DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* ************
 TTGG            S    4      BASIC TASK SUMMARY FILE NAME SPECIFIER

 **** NOTES ****
 THIS VARIABLE REPLACES THE POSTIONS 5-8 IN THE FILE NAME 'BTSMttgg.XDB'

**********************************************************
```

```
*******************************************************

 FILE NAME: BTSMttgg.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: BTSREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 1014
 PAGE SIZE   : 1024
 RECORD NAME : BTSREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     BTSKEY       1        7     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 BTSKEY          S   7           1       COMPONENT ID
 BTSUMI          I   2           9       UNIT OF MEASURE ID
 BTSTRD          I   2           11      TRADE INDEX
 BTSCLS          I   2           13      CLASSIFICATION ID
 BTSWPM          S   (6)1        15      WORK PERFORMANCE METHOD
 BTSNYR          I   2           23      NUMBER OF YEARS
 BTSDES          S   30          25      COMPONENT DESCRIPTION
 BTSDTA          R   (80,3)      55      ARRAY OF DATA(LABOR,MATERIAL,EQUIPMENT)
 BTSLAB          R   (80)4       55      LABOR HOURS ARRAY (80 YEARS)
 BTSMAT          R   (80)4       374     MATERIALS HOURS ARRAY (80 YEARS)
 BTSEQP          R   (80)4       694     EQUIPMENT HOURS ARRAY (80 YEARS)

 **** NOTES ****
 THE LAST TREE FIELDS ARE EQUIVALENCED TO 'BTSDTA'

*******************************************************


*******************************************************

 FILE NAME: CLASLST.DAT
 FILE TYPE: TEXT
 NUMBER OF KEYS:  1
 NUMBER OF LINES: 1

 **** FILE INFORMATION ****
  LINE     FORMAT     FIELDS
 *****   **********  **********
   1       I2,A30       INDEX,DESCR

 ***** FIELD INFORMATION *****
 FIELD          TYPE LENGTH  DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* ************
 INDEX           S    2      CLASIFICATION ID ** KEY **
 DESCR           S    30     CLASIFICATION DESCRIPTION

*******************************************************
```

```
*******************************************************

 FILE NAME: CTODffff.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: CTREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 28
 PAGE SIZE   : 512
 RECORD NAME : CTREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     CTTSKX       1        7     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 CTTSKX          S   7          1       CACES SUMMARY OR COMPONENT
 CTFLGS          S   (4)1       9       INDICATING ENTRY BY 'C'OMPUTER OR 'U'SE
 CTQTY           R   4          13      QUANTITY
 CTIDTE          I   2          17      INSTALLATION DATE
 CTLDTE          I   2          19      LAST PERFORMED DATE
 CTNDTE          I   2          21      NEXT PERFORMED DATE

 **** NOTES ****
 THE FILE NAME IS BUILT USING THE FIELD 'FGXSEQ' IN THE FILE 'FACILITY.XDB'.
 THIS FIELD REPLACES THE 'ffff' IN THE FILE NAME.

*******************************************************


*******************************************************

 FILE NAME: DES-BTSM.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 34
 PAGE SIZE   : 512
 RECORD NAME : DESDAT

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     KEY          1        4     Y   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 TREEID          S   2          1       TREE TABLE ID
 GRPID           S   2          3       BASIC TASK TABLE SUMMARY ID
 DESC            S   30         5       BASIC TASK SUMMARY TABLE NAME

 **** NOTES ****
 THE FIELDS 'TREEID' AND 'GRPID' ARE COMBINED TO FORM THE KEY FIELD


*******************************************************
```

```
**********************************************************

 FILE NAME: DES-TASK.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 34
 PAGE SIZE   : 512
 RECORD NAME : DESDAT

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     KEY         1        4     Y   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 TREEID          S   2           1       TREE TABLE ID
 GRPID           S   2           3       BASIC TASK TABLE ID
 DESC            S   30          5       BASIC TASK TABLE NAME

 **** NOTES ****
 THE FIELDS 'TREEID' AND 'GRPID' ARE COMBINED TO FORM THE KEY FIELD.


**********************************************************


**********************************************************

 FILE NAME: DES-TRWD.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 34
 PAGE SIZE   : 512
 RECORD NAME : DESDAT

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     KEY         1        4     Y   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 TREEID          S   2           1       TREE TABLE ID
 GRPID           S   2           3       BASIC TASK TABLE ID
 DESC            S   30          5       TREE WIDE TABLE NAME

 **** NOTES ****
 THE FIELDS 'TREEID' AND 'GRPID' ARE COMBINED TO FORM THE KEY FIELD


**********************************************************
```

```
*******************************************************

 FILE NAME: DRSPE.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: DRREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 14
 PAGE SIZE   : 512
 RECORD NAME : DRREC

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     DIRSPE      1         8     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 DIRSPE          S   8           1       DIRECTORY
 DRIVES          S   1           9       DRIVE


*******************************************************


*******************************************************

 FILE NAME: EQC-TAB.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: EQCREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 50
 PAGE SIZE   : 512
 RECORD NAME : EQCREC

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     EQCCDE      1         2     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 EQCCDE          S   2           1       EQUIPMENT ID
 EQCDES          S   40          3       EQUIPMENT DESCRIPTION
 EQCCOST         R   4           43      DOLLARS PER HOUR

*******************************************************
```

```
*******************************************************

 FILE NAME: F4C-YEAR.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: FYREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 34
 PAGE SIZE   : 512
 RECORD NAME : FXREC

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     FXF4CB      1         7     Y   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 FXF4CB          S   7           1       BEGINNING F4C CODE
 FXF4CE          S   7           9       ENDING F4C CODE
 FXTEID          S   2           17      TREE ID TABLE
 FXBTID          S   2           19      BASIC TASK ID TABLE
 FXSMID          S   2           21      SUMMARY ID TABLE
 FXPTID          S   2           23      PARTIAL SUMMARY ID TABLE
 FXUNID          S   2           25      UNIT COST ID TABLE
 FXBYER          S   4           27      BEGINING YEAR
 FXEYER          S   4           31      ENDING YEAR

*******************************************************
*******************************************************

 FILE NAME: FACTAB.DAT
 FILE TYPE: TEXT
 NUMBER OF LINES: 1

 FORMAT USED : A80

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 FGID            S   9           1       FACILITY ID
 FGSUBI          S   2           10      SUB INSTALLATION ID
 FGAREA          S   2           12      AREA ID
 FGDESC          S   30          14      FACILITY DESCRIPTION
 FGF4C           S   7           29      F4C CODE
 FGNUM           I   2           36      NUMBER OF FACILITIES
 FGZONE          I   2           39      TRAVEL ZONE
 FGMTH           I   2           41      WORK PERFORMANCE METHOD
 FGSCM           I   2           43      SPECIAL CONDITION MULTIPLIER ID
 FGCHNG          S   8           45      LAST CHANGED DATE
 FGLCAL          S   8           53      LAST CALCULATION DATE
 FGFUND          S   10          61      FACILITY FUNDING PROFILE
 FGCAL           I   2           63      CALCULATION MODELING ID
 FGSQFT          I   4           64      FLOOR AREA (SQUARE FEET)
 FGCYR           I   2           73      CONSTRUCTION YEAR
 FGSDSP          S   8           77      SCHEDUALED DISPOSAL DATE

 **** NOTES ****
 USED TO BUILD THE 'FACILITY.XDB' FILE IN BATCH MODE.
*******************************************************
```

****

FILE NAME: FACILITY.XDB
FILE TYPE: BTRIEVE
INCLUDE FILE: FAREC.INC
NUMBER OF KEYS: 3
RECORD SIZE : 120
PAGE SIZE : 1024
RECORD NAME : FGREC

**** KEY INFORMATION ****

KEYNO	FIELD	POSITION	LENGTH	DUP	MOD	TYPE
0	FGID	1	9	N	N	S
1	FGXSEQ	11	4	N	N	S
2	FGF4C	49	7	Y	Y	S

***** RECORD INFORMATION *****

FIELD NAME	TYPE	LENGTH (BYTES)	POSITION	DESCRIPTION
FGID	S	9	1	FACILITY ID
FGXSEQ	S	4	11	FACILITY SEQUENCE NUMBER
FGDESC	S	30	15	FACILITY DESCRIPTION
FGSUBI	S	2	45	SUB INSTALLATION ID
FGAREA	S	2	47	AREA ID
FGF4C	S	7	49	F4C CODE
FGNUM	I	2	57	NUMBER OF FACILITIES
FGZONE	I	2	59	TRAVEL ZONE
FGSQFT	I	4	61	FLOOR AREA (SQUARE FEET)
FGCYR	I	2	65	CONSTRUCTION YEAR
FGSDSP	S	8	67	SCHEDULED DISPOSAL DATE
FGFUND	S	2	75	FACILITY FUNDING PROFILE
MODSYS	S	1	77	COMPONENTS ENTERED
BFAPCT	I	2	79	PERCENTAGE OF BASE FACILITY
FGSCM	I	2	85	SPECIAL CONDITION MULTIPLIER ID
FGMTH	I	2	87	WORK PERFORMANCE METHOD
FGCAL	I	2	89	CALCULATION MODELING ID
FGCHNG	S	8	91	LAST CHANGED DATE
FGLCAL	S	8	99	LAST CALCULATION DATE
FGDIR	S	2	109	DIRECTORY LOCATION OF FACILITY FILES
BSFGID	S	9	111	BASE FACILITY ID

****

```
**********************************************************

 FILE NAME: FFPROF.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: FFPROF.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 90
 PAGE SIZE   : 512
 RECORD NAME : FFREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     FFKEY        1        2     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 FFKEY           S   2           1       FACILITY FUNDING PROFILE ID
 FFDSCR          S   40          3       PROFILE DESCRIPTION
 FFLAP           S   (5)2        43      IN-HOUSE LABOR APPROPRIATION ID
 FFLPR           I   (5)2        53      IN-HOUSE LABOR PERCENTAGE
 FFEAP           S   (5)2        63      IN-HOUSE EQUIPMENT APP. ID
 FFEPR           I   (5)2        73      IN-HOUSE EQUIPMENT PERCENTAGE
 FFLNO           I   4           83      NUMBER OF LABOR ACCOUNTS
 FFENO           I   4           87      NUMBER OF EQUIPMENT ACCOUNTS

**********************************************************


**********************************************************

 FILE NAME: INSTINFO.DAT
 FILE TYPE: TEXT
 NUMBER OF LINES: 6

 **** FILE INFORMATION ****
 LINE     FORMAT       FIELDS
 *****  ***********  ***********
  1        A30         INSTNAM
  2     1X,A4,1X,A4    BEGYR,ENDYR
  3       3F5.2        MATADJ,MATTAF,RMFTAF
  4        A2          ORGID
  5        A2          MAXLIN
  6        A1          VDRIVE

 ***** FIELD INFORMATION *****
 FIELD          TYPE LENGTH  DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* ************
 INSTNAM         S     30     INSTALLATION NAME
 BEGYR           S      4     BEGINNING REPORT YEAR
 ENDYR           S      4     ENDING REPORT YEAR
 MATADJ          R      4     MATERIAL LOCATION ADJUSTMENT FACTOR
 MATTAF          R      4     MATERIAL TIME ADJUSTMENT FACTOR
 RMFTAF          R      4     RMF TIME ADJUSTMENT FACTOR
 ORGID           S      2     ORGANIZATION ID
 MAXLIN          S      2     MAXIMUM NUMBER OF LINES PER PAGE
 VDRIVE          S      1     DRIVE DESIGNATION OF VIRTUAL DISK, C=NONE

**********************************************************
```

****

FILE NAME: ORGFGC.XDB
FILE TYPE: BTRIEVE
INCLUDE FILE: ***NONE***
NUMBER OF KEYS: 2
RECORD SIZE : 52
PAGE SIZE : 512
RECORD NAME : ORGREC

**** KEY INFORMATION ****

KEYNO	FIELD	POSITION	LENGTH	DUP	MOD	TYPE
0	ORGKEY	1	4	N	N	S
1	MACOMID	43	2	Y	Y	S

***** RECORD INFORMATION *****

FIELD NAME	TYPE	LENGTH (BYTES)	POSITION	DESCRIPTION
ORGKEY	S	4	1	ORGANIZATION ID
ORGCDE	S	2	5	ORGANIZATION CODE
INSTID	S	2	7	INSTALLATION ID
ORGDESC	S	30	13	ORG. DESCRIPTION
MACOMID	S	2	43	MACOM ID
RELCDE	S	6	45	RELATION CODE
SUBCDE	S	2	51	SUB-INSTALLATION CODE

****

****

FILE NAME: PMCOMP.XDB
FILE TYPE: BTRIEVE
INCLUDE FILE: *** NONE ***
NUMBER OF KEYS: 3
RECORD SIZE : 64
PAGE SIZE : 512
RECORD NAME : COMPDT

**** KEY INFORMATION ****

KEYNO	FIELD	POSITION	LENGTH	DUP	MOD	TYPE
0	STF4C	1	8	Y	Y	S
1	CASCES	17	8	Y	Y	S
2	IFSNUM	25	8	Y	Y	S

***** RECORD INFORMATION *****

FIELD NAME	TYPE	LENGTH (BYTES)	POSITION	DESCRIPTION
STF4C	S	8	1	STARTING F4C CODE
ENDF4C	S	8	9	ENDING F4C CODE
CASCES	S	8	17	TASK/COMPONENT ID
IFSNUM	S	8	25	IFS NUMBER
PMNUM	S	2	33	NUMBER OF THE CURRENT PREDICTION MODEL
LCALDTE	S	8	35	LAST CALCULATION DATE
LMRGDTE	S	8	43	LAST MERGE DATE
TOTPM	S	2	51	TOTAL NUMBER OF ALLOWABLE PRED. MODELS
ALLOWPM	S	(6)2	53	LIST OF THE ALLOWABLE PRED. MODELS

****

```
*******************************************************

 FILE NAME: PMDEF.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 42
 PAGE SIZE   : 512
 RECORD NAME : PMDDAT

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     PMNUM        1         2     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 PMID            S   2           1       PREDICTION MODEL ID
 PMDEF           S   40          3       PREDICTION MODEL DEFINITION

*******************************************************

*******************************************************

 FILE NAME: PMF4C.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 48
 PAGE SIZE   : 512
 RECORD NAME : F4CDAT

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     STF4C        1         8     N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 STF4C           S   8           1       STARTING F4C CODE
 ENDF4C          S   8           9       ENDING F4C CODE
 PMNUM           S   2           17      NUMBER OF THE CURRENT PREDICTION MODEL
 LCALDTE         S   8           19      LAST CALCULATION DATE
 LMGRDTE         S   8           27      LAST MERGE DATE
 TOTPM           S   2           35      TOTAL NUMBER OF ALLOWABLE PRED. MODELS
 ALLOWPM         S   2(6)        37      LIST OF THE ALLOWABLE PRED. MODELS

*******************************************************
```

```
**********************************************************

 FILE NAME: RMF-FACT.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 336
 PAGE SIZE   : 512
 RECORD NAME : RMFREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     STAMS        1         7     N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME              (BYTES)
 *****          **** ******* *********  ************
 STAMS           S   7           1       STARTING AMS CODE
 ENDAMS          S   7           9       ENDING AMS CODE
 RMCOST          R   (80)4       17      ARRAY OF COST FACTORS (80 YEARS)

**********************************************************
```

```
**********************************************************

 FILE NAME: ROccapam.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: RSREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 616
 PAGE SIZE   : 1024
 RECORD NAME : RSREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     RSRTSK       1         7     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 RSRTSK          S   8           1       COMPONENT ID
 RSRTRD          I   2           9       TRADE INDEX
 RSRWPM          I   2           11      WORK PERFORMANCE METHOD
 RSRBFY          I   2           13      BEGINNING YEAR
 RSRNFY          I   2           15      NUMBER OF YEARS
 RSROCC          R   (10)4       17      NUMBER OF OCCURRENCES (10 YEARS)
 RSRTOT          R   (10)4       57      TOTAL COSTS (10 YEARS)
 RSRHRS          R   (3,10)      137     ARRAY OF HOURS (10 YEARS)
 RSRDLR          R   (3,10)      377     ARRAY OF COSTS (10 YEARS)

 **** NOTES ****
 BOTH TWO DIMENTIONAL ARRAYS (RSRHRS & RSRDLR) ARE DIVIDED AS FOLLOWS:
   (1,I) = LABOR
   (2,I) = EQUIPMENT
   (3,I) = MATERIALS

 THE FILE NAME IS CONSTRUCTED AS FOLLOWS:
   oc = ORGANIZATION CODE
   ap = APPROPRIATION ID
   am = AMS ID

**********************************************************
```

```
**********************************************************

 FILE NAME: RSMYffff.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: RSREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 340
 PAGE SIZE   : 1024
 RECORD NAME : RSREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     RSRTSK       1         7     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 RSRTSK          S   8           1       COMPONENT ID
 RSRTRD          I   2           9       TRADE INDEX
 RSRWPM          I   2           11      WORK PERFORMANCE METHOD
 RSRBFY          I   2           13      BEGINNING YEAR
 RSRNFY          I   2           15      NUMBER OF YEARS
 RSROCC          R   (10)4       17      NUMBER OF OCCURRENCES (10 YEARS)
 RSRTOT          R   (10)4       57      TOTAL COSTS (10 YEARS)
 RSRHRS          R   (3,10)      97      ARRAY OF HOURS (10 YEARS)
 RSRDLR          R   (3,10)      217     ARRAY OF COSTS (10 YEARS)

 **** NOTES ****
 BOTH TWO DIMENSIONAL ARRAYS (RSRHRS & RSRDLR) ARE DIVIDED AS FOLLOWS:
   (1,I) = LABOR
   (2,I) = EQUIPMENT
   (3,I) = MATERIALS

 THE 'ffff' IN THE FILE NAME IS REPLACED BY THE FACILITY SEQUENCE NUMBER FROM
 FILE 'FACILITY.XDB', FIELD NAME IS 'FGXSEQ'

**********************************************************
```

```
**********************************************************

 FILE NAME: RSMTTOTL.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: RSREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 340
 PAGE SIZE   : 1024
 RECORD N  E : RSREC

 **** KE   FORMATION ****
 KEYNO  F  LD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     RSRTSK      1         7     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 RSRTSK          S   8           1       COMPONENT ID
 RSRTRD          I   2           9       TRADE INDEX
 RSRWPM          I   2           11      WORK PERFORMANCE METHOD
 RSRBFY          I   2           13      BEGINNING YEAR
 RSRNFY          I   2           15      NUMBER OF YEARS
 RSROCC          R   (10)4       17      NUMBER OF OCCURRENCES (10 YEARS)
 RSRTOT          R   (10)4       57      TOTAL COSTS (10 YEARS)
 RSRHRS          R   (3,10)      97      ARRAY OF HOURS (10 YEARS)
 RSRDLR          R   (3,10)      217     ARRAY OF COSTS (10 YEARS)

 **** NOTES ****
 BOTH TWO DIMENTIONAL ARRAYS (RSRHRS & RSRDLR) ARE DIVIDED AS FOLLOWS:
   (1,I) = LABOR
   (2,I) = EQUIPMENT
   (3,I) = MATERIALS

**********************************************************
```

```
**********************************************************

 FILE NAME: RSXXffff.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: RS120.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 340
 PAGE SIZE   : 1024
 RECORD NAME : RSREC

 **** KEY INFORMATION ****
 KEYNO  FIELD    POSITION  LENGTH  DUP MOD TYPE
 *****  *****    ********  ******  *** *** ****
  0     RSRTSK      1         8     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 RSRTSK          S   8           1       COMPONENT ID
 RSRTRD          I   2           9       TRADE INDEX
 RSRWPM          I   2           11      WORK PERFORMANCE METHOD
 RSRBFY          I   2           13      BEGINNING YEAR
 RSRNFY          I   2           15      NUMBER OF YEARS
 RSROCC          R   (10)4       17      NUMBER OF OCCURRENCES (10 YEARS)
 RSRTOT          R   (10)4       57      TOTAL COSTS (10 YEARS)
 RSRHRS          R   (3,10)      97      ARRAY OF HOURS (10 YEARS)
 RSRDLR          R   (3,10)      21      ARRAY OF COSTS (10 YEARS)

 **** NOTES ****
 BOTH TWO DIMENTIONAL ARRAYS (RSRHRS & RSRDLR) ARE DIVIDED AS FOLLOWS:
   (1,I) = LABOR
   (2,I) = EQUIPMENT
   (3,I) = MATERIALS

 THE 'ffff' IN THE FILE NAME IS REPLACED BY THE FACILITY SEQUENCE NUMBER FROM
 FILE 'FACILITY.XDB', FIELD NAME IS 'FGXSEQ'


**********************************************************
```

```
**********************************************************

 FILE NAME: SCMDEF.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 32
 PAGE SIZE   : 512
 RECORD NAME : SCMDAT

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     SCMID        1        2     N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 SCMID           S   2           1       SPECIAL CONDITION MULTIPLIER ID
 SCMDEF          S   30          3       SCM DEFINITION

**********************************************************

**********************************************************

 FILE NAME: SCMDEF.DAT
 FILE TYPE: TEXT

 FORMAT USED: A2,A30

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 CODE            S   2           1       SPECIAL CONDITION MULTIPLIER ID
 DESC            S   30          3       SCM DEFINITION

**********************************************************
```

```
**********************************************************

 FILE NAME: SCMIDxx.DAT
 FILE TYPE: TEXT
 NUMBER OF LINES: 20

 **** FILE INFORMATION ****
 LINE     FORMAT       FIELDS
 *****  ********     ******
  1         I5         NUMBER OF DATA LINES(19)
  2-20  A7,3X,F10.2   COMPONENT ID, CALCULATION TOTALS

 ***** RECORD INFORMATION *****
 LINE     COMPONENT  CALCULATION
 *****  *********** ************
  2        0000000   F(9)*F(13)
  3        0300000   F(1)*F(2)*F(5)*F(6)*F(7)*F(8)
  4        0400000   F(1)*F(2)*F(5)*F(6)*F(8)*F(11)
  5        0410000   F(15)
  6        0415400   F(4)
  7        0415500   F(4)
  8        0415800   F(4)
  9        0415900   F(4)
 10        0415A00   F(4)
 11        0415E00   F(14)
 12        0415F00   F(14)
 13        0415G00   F(14)
 14        0415H00   F(14)
 15        0420000   F(14)
 16        0430000   F(14)
 17        0500000   F(11)
 18        0530000   F(12)
 19        0540000   F(12)
 20        0600000   F(11)

 **** NOTES ****
 THE 'xx' IS REPLACED BY THE SPECIAL CONDITION MULTIPLIER ID

**********************************************************
```

```
*******************************************************

 FILE NAME: STDREP.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 4
 PAGE SIZE   : 512
 RECORD NAME : STDREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     STDKEY       1        4     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 STDAPR          S   2          1       APPROPRIATION ID
 STDAMS          S   2          3       AMS CODE

 **** NOTES ****
  THE FIELDS 'STDAPR' AND 'STDAMS' ARE COMBINED TO FORM THE KEY FIELD

*******************************************************

*******************************************************

 FILE NAME: SUB_TAB.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: ***NONE***
 NUMBER OF KEYS:  1
 RECORD SIZE : 32
 PAGE SIZE   : 512
 RECORD NAME : ARDAT

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     CODE         1        2     N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 CODE            S   2          1       SUB-INSTALLATION ID
 DEF             S   30         3       SUB-INSTALLATION DESCRIPTION

*******************************************************
```

```
*****************************************************

 FILE NAME: TASKttgg.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: BTIREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 90
 PAGE SIZE   : 512
 RECORD NAME : BTIREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     BTIKEY       1         7     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 BTIKEY          S   7           1       TASK ID
 BTIDES          S   40          9       TASK DESCRIPTION
 BTIUM           I   2           49      UNIT OF MEASURE ID
 BTITRD          I   2           53      TRADE INDEX
 BTIDEF          I   2           55      TASK CLASSIFICATION
 BTIHGH          R   4           57      TASK HIGH FREQUENCY
 BTIAVE          R   4           61      TASK AVERAGE FREQUENCY
 BTILOW          R   4           65      TASK LOW FREQUENCY
 BTILAB          R   4           69      LABOR HOURS
 BTIMAT          R   4           73      MATERIAL COSTS
 BTIEQP          R   4           77      EQUIPMENT COSTS
 TWPMTH          S   (6)1        80      TASK WORK PERFORMANCE METHOD
 BTIEID          S   2           87      EQUIPMENT ID

 **** NOTES ****
 FILE NAME IS BUILT USING 'FXTEID' AND 'FXBTID' FIELDS FROM THE FILE
 'F4C-YEAR.XDB'.  FXTEID REPLACES 'tt' AND FXBTID REPLACES 'gg'.

*****************************************************
```

****

FILE NAME: TRAVTIME.DAT
FILE TYPE: TEXT
NUMBER OF KEYS: 1

**** KEY INFORMATION ****

FIELD	POSITION	LENGTH
*****	********	******
ZONE	1	2

***** RECORD INFORMATION *****

FIELD NAME	TYPE	LENGTH (BYTES)	POSITION	DESCRIPTION
*****	****	*******	********	**********
ZONE	S	2	1	TRAVEL ZONE
TIME	S	5	3	TRAVEL TIME

****

****

FILE NAME: TRDCOSTS.DAT
FILE TYPE: TEXT
INCLUDE FILE:
NUMBER OF KEYS: 0
RECORD SIZE :
PAGE SIZE :
RECORD NAME :

**** KEY INFORMATION ****

FIELD	POSITION	LENGTH
*****	********	******
TI	1	2

FORMAT USED: A2,A40,A2,A10,A10,A10,A10

***** RECORD INFORMATION *****

FIELD NAME	TYPE	LENGTH (BYTES)	POSITION	DESCRIPTION
*****	****	*******	********	**********
TI	S	2	1	TRADE INDEX
DESC	S	40	3	DESCRIPTION
SHPCD	S	2	43	SHOP CODE
ILABC	S	10	52	IN-HOUSE LABOR
CLABC	S	10	61	CONTRACT LABOR
IEQC	S	10	70	IN-HOUSE EQUIPMENT
CEQC	S	10	79	CONTRACT EQUIPMENT

****

****

FILE NAME: TREEffff.DAT
FILE TYPE: BINARY, DI
INCLUDE FILE: TSKREC.INC
NUMBER OF KEYS: 1
RECORD SIZE : 60
RECORD NAME : TSKREC

**** KEY INFORMATION ****
-- RECORD NUMBER ---

***** RECORD INFORMATION *****

FIELD NAME	TYPE	LENGTH (BYTES)	POSITION	DESCRIPTION
*****	****	*******	*********	**********
TSKIDX	S	7	1	TREE ID (TASK/COMPONENT)
TSKDES	S	40	9	TASK/COMPONENT DESCRIPTION
TSKSMY	S	7	49	SUMMARY TASK ID
DWNPTR	I	2	57	DOWN POINTER (NEXT LOGICAL RECORD)
BRNCNT	I	2	59	BRANCH COUNT (TOTAL NUMBER IN BRANCH)

**** NOTES ****

THE 'ffff' IN THE FILE NAME IS REPLACED BY THE FACILITY SEQUENCE NUMBER FROM FILE 'FACILITY.XDB', FIELD NAME IS 'FGXSEQ'

****

****

FILE NAME: TRWDaa--.DAT
FILE TYPE: BINARY, DI
INCLUDE FILE: TSKREC.INC
NUMBER OF KEYS: 1
RECORD SIZE : 60
RECORD NAME : TSKREC

**** KEY INFORMATION ****
-- RECORD NUMBER ---

***** RECORD INFORMATION *****

FIELD NAME	TYPE	LENGTH (BYTES)	POSITION	DESCRIPTION
*****	****	*******	*********	**********
TSKIDX	S	7	1	TREE ID (TASK/COMPONENT)
TSKDES	S	40	9	TASK/COMPONENT DESCRIPTION
TSKSMY	S	7	49	SUMMARY TASK ID
DWNPTR	I	2	57	DOWN POINTER (NEXT LOGICAL RECORD)
BRNCNT	I	2	59	BRANCH COUNT (TOTAL NUMBER IN BRANCH)

**** NOTES ****

THE 'ffff' IN THE FILE NAME IS REPLACED BY THE FACILITY SEQUENCE NUMBER FROM FILE 'FACILITY.XDB', FIELD NAME IS 'FGXSEQ'

****

```
**********************************************************

 FILE NAME: TRWDaa--.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: REFREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 12
 PAGE SIZE   : 512
 RECORD NAME : REFREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     REFKEY       1         7     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 REFKEY          S   8          1       CACES NUMBER
 NODEPOS         I   2          9       RECORD NUMBER OF KEY IN TREE FILE

 **** NOTES ****
  FIELD 'FXTEID' FROM FILE 'F4C-YEAR.XDB' REPLACES 'aa' IN FILE NAME.


**********************************************************


**********************************************************

 FILE NAME: UNCDSC.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: UNCDSC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 52
 PAGE SIZE   : 512
 RECORD NAME : UNCRC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     UNTID        1         2     N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME                (BYTES)
 *****          **** ******* *********  ************
 UNTID           S   2          1       UNIT COST TABLE ID
 DESP            S   50         3       DESCRIPTION


**********************************************************
```

```
**********************************************************

 FILE NAME: UNC-FACT.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: UXREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 360
 PAGE SIZE   : 512
 RECORD NAME : UNCREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     UNCID        1        2     N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 UNCID           S   2          1       UNIT COST TABLE ID
 UNCSPM          R   4          5       UNIT COST SPECIAL CONDITION MULTIPLIER
 UNARM           R   4          9       ANNUAL RECURRING MAINTAINANCE FACTOR
 UNCOST          R   4(80)      13      UNIT COST FACTORS FOR 80 YEARS

**********************************************************


**********************************************************

 FILE NAME: URR.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: URRINC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 90
 PAGE SIZE   : 512
 RECORD NAME : URRREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     AMSCOD       1        10    N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 AMSCOD          S   10         1       AMS CODE
 URRKSF          R   4(10)      11      URR KILO SQUARE FOOT
 URRKDO          R   4(10)      51      URR THOUSAND DOLLAR

**********************************************************
```

```
**********************************************************

 FILE NAME: URRAPR.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: NONE
 NUMBER OF KEYS:  1
 RECORD SIZE : 50
 PAGE SIZE   : 512
 RECORD NAME : APRMUL

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     AMSCOD       1         6    N   Y    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* *********  ************
 APRID           S   6           1      APPROPRIATION ID
 MULPLY          R   4(10)       7      MULTIPLIER

**********************************************************

**********************************************************

 FILE NAME: VALLIST.DAT
 FILE TYPE: TEXT
 NUMBER OF KEYS:  1

 **** FILE INFORMATION ****
 FORMAT    FIELDS
 *****     ********
 I2,A30    INDEX,DESCR

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  DESCRIPTION
 NAME               (BYTES)
 *****          **** ******* ************
 INDEX           I     2     UNIT OF MEASURE ID ** KEY **
 DESCR           S    30     DESCRIPTION

**********************************************************
```

```
*******************************************************

 FILE NAME: WP-DESC.DAT
 FILE TYPE: TEXT
 NUMBER OF LINES:  6

 **** FILE INFORMATION ****
 LINE   FORMAT          FIELDS
 *****  *******        ********
  1-6    A30            MTHDES(N)

 ***** FIELD INFORMATION *****
 FIELD          TYPE LENGTH   DESCRIPTION
 NAME                (BYTES)
 *****          **** *******  ************
 MTHDES          S     30     WORK PERFORMANCE METHOD DESCRIPTION

*******************************************************

*******************************************************

 FILE NAME: WP-PASS.DAT
 FILE TYPE: TEXT
 NUMBER OF LINES:  1

 **** FILE INFORMATION ****
 LINE   FORMAT          FIELDS
 *****  *******        ********
  1      A4              TTGG

 ***** FIELD INFORMATION *****
 FIELD          TYPE LENGTH   DESCRIPTION
 NAME                (BYTES)
 *****          **** *******  ************
 TTGG            S      4     BASIC TASK FILE NAME SPECIFIER


 **** NOTES ****
 THIS VARIABLE REPLACES THE POSITIONS 5-8 IN THE FIELD NAME 'TASKttgg.XDB'.

*******************************************************
```

```
**********************************************************

 FILE NAME: ZOsixf4c.XDB
 FILE TYPE: BTRIEVE
 INCLUDE FILE: RXREC.INC
 NUMBER OF KEYS:  1
 RECORD SIZE : 500
 PAGE SIZE   : 1024
 RECORD NAME : RXREC

 **** KEY INFORMATION ****
 KEYNO  FIELD     POSITION  LENGTH  DUP MOD TYPE
 *****  *****     ********  ******  *** *** ****
  0     TASKID       1        8     N   N    S

 ***** RECORD INFORMATION *****
 FIELD          TYPE LENGTH  POSITION   DESCRIPTION
 NAME               (BYTES)
 ***            **** ******* *********  ************
 TASKID          S   8          1       COMPONENT ID
 NOYEAR          I   2          9       NUMBER OF DECADES BEEN CALCULATED
 RX120           R   (120)4     11      120 YEARS DATA
 RXTOT           R   4          491     120 YEARS SUMMARY

 **** NOTES ****
   TASKID(1:7) = TASK ID
   TASKID(8:8) = RECORD INDEX
                 1 : Labor     Hours
                 2 : Material  Cost
                 3 : Equipment Hours
                 4 : Total     Costs

 THE FILE NAME IS CONSTRUCTED AS FOLLOWS:
   sixf4c : six right most F4C code
**********************************************************
```

6 STANDARD PROGRAMMING PACKAGES

These programming packages are used by the MRPM system:

1. MSFORTRAN 3.31
2. BTRIEVE
3. MS CHART

Each package has its own printed documentation.

7 MAINTENANCE AND OPERATIONS PROCEDURES

To ensure system uniformity, programmers should follow these instructions:

1. Always get source codes from and return them to the coordinator; do not update the execution file in the main machine. Only the coordinator is allowed to write into the main machine.

2. Test the program as you work on it, and make sure it is 100 percent error free. Write down all testing procedures for future reference.

3. The first line of each screen should be a header that contains the exact words of the selection menu.

4. Always show the computer status on the screen: 'COMMAND MODE', 'EDIT MODE','ADD MODE'; this helps users know where they are.

5. When the user presses 'F8' to delete records, always ask to make sure the data should be deleted.

6. Use the bottom line for standard functions. Show only functions that can be used in this mode.

7. On reports and lists to the printer, be sure to print the last page.

8. Use the 'F6 BEGIN' key to start the processing on all functions.

9. Programs must return to the previous screen after successful execution.

10. Use the standard error message subroutine "ERRMSG" to handle all the pop-up window style error messages.

11. Always make a source code backup before working on it. Always save the final program on a diskette in your diskette file. Write the date on the diskette.

12. There is no 'STOP' command in the program.

13. Increase labels in increments of 100.

Table 4 is an example of a well-documented program.

Table 4

Example of a Well Documented Program

```
      PROGRAM RECFACZ
* ==> this program recovers the general facility information
*     stored in FACILITY.XDB from subdirectories
      IMPLICIT    INTEGER*2 (A-Z)
      CHARACTER   SDIR*2,EDIR*2,BUFFER*30
      INTEGER*2   FILE1(64),FILE2(64),NSDIR,NEDIR,DIR
      INTEGER*2   CODE,OP
      CHARACTER   KEY*9
      CHARACTER   PATH*18,SDRIVE*1,NEWDIR*2
      LOGICAL*2   THERE,OPEND,SUCESS
      COMMON /COLORS/ COLOR
* ==> include file for error message
$INCLUDE:'BTEROR.INC'
* ==> include file for FACILITY.XDB
$INCLUDE:'FAREC.INC'
* ==> include files for color table
$INCLUDE:'COLOR1.INC'
$INCLUDE:'COLOR2.INC'
      ROUTIN = 'REC-FAC'
* ==> inialize the error message subroutine
      CALL ERRMSG(-1,ROUTIN,'INITAL')
* ==> install function keys
      CALL INFKEY
* ==> Make sure Record Manager is loaded
      IF (BTREEX(0).NE.0) THEN
        INFOE(1)='Record Manager is not Loaded'
        CALL ERRMSG(99,ROUTIN,' ')
        GOTO 999
      ENDIF
* ==> set up for screen
100   CALL SCROLL (00,00,24,80,00,COLOR(2))
      CALL FLINE
      CALL FKEYS (1,1,1,1,1,1,1,1,1,2)
      CALL BOX(1,1,3,80,2,COLOR(2))
      CALL BOX(4,1,24,80,2,COLOR(2))
      CALL WT (2,25,' REBUILD FACILITY.XDB FROM SUBS    ',31,COLOR(14))
      CALL WT (10,15,'Starting Subdirectory :',23,COLOR(14))
      CALL WT (12,15,'Ending   Subdirectory :',23,COLOR(14))
* ==> read screen
200   CODE = 5
      CALL RT (10,39,SDIR,2,CODE,COLOR(8),COLOR(15))
      IF (CODE.EQ.68) GOTO 999
      IF (CODE.EQ.80.OR.CODE.EQ.13) GOTO 300
      GOTO 200
300   CODE = 5
      CALL RT (12,39,EDIR,2,CODE,COLOR(8),COLOR(15))
      IF (CODE.EQ.68) GOTO 999
      IF (CODE.EQ.64) GOTO 400
      IF (CODE.EQ.72) GOTO 200
      IF (CODE.EQ.13) THEN
        CALL FLINE
        CALL FKEYS (1,1,1,1,1,2,1,1,1,2)
      ENDIF
      GOTO 300
400   CONTINUE
* ==> create FACTEMP.XDB in current directory
```

```
      CALL CRFAC(FILE1,SUCESS)
* ==> if not created sucessfully terminate the program
      IF (.NOT.SUCESS) GOTO 999
* ==> open FACTEMP.XDB in current directory
      CALL BTREE (FILE1,0,0,'FACTEMP.XDB',9,FGREC,120,STS)
      IF (STS.NE.0) THEN
         INFOE(1) = 'ERROR OPEN'
         CALL ERRMSG(STS,ROUTIN,'DF..98')
         GOTO 999
      ENDIF
      READ(SDIR,'(I2)') NSDIR
      READ(EDIR,'(I2)') NEDIR
      DIR = NSDIR
500   CONTINUE
      OPEND=.FALSE.
      WRITE(NEWDIR,'(I2.2)') DIR
      CALL FINDDR(NEWDIR,SDRIVE)
* ==> set up filename for subdirectory general facility file
      PATH='E:\\01\\FACILITY.XDB'c
      PATH(1:1)=SDRIVE
      PATH(4:5)=NEWDIR
* ==> open FACILITY.XDB in subdirectory
      CALL BTREE (FILE2,0,0,PATH,18,FGREC,120,STS)
      IF (STS.eq.12) THEN
         INFOE(1) = 'FACILITY.XDB not found in Subdirectory'
         INFOE(2) = PATH
         INFOA(1) = 'Please check files in Subdirectory'
         CALL ERRMSG(99,ROUTIN,' ')
         GOTO 800
      ENDIF
      IF (STS.NE.0)  THEN
         FILNAM = 'FACILITY.XDB'
         INFOE(1) = 'Error in opening file'
         CALL ERRMSG(STS,ROUTIN,'OPEN02')
         GOTO 800
      ENDIF
      OPEND=.TRUE.
* ==> write processing information to screen
      BUFFER = 'RECOVER FACILITY IN  E:\'
      BUFFER(22:22) = SDRIVE
      WRITE(BUFFER(25:26),'(I2.2)') DIR
      CALL WT (20,15,BUFFER,30,COLOR(14))
* ==> loop for copy facility records
      OP = 12
600   CONTINUE
      CALL BTREE (FILE2,OP,0,KEY,9,FGREC,120,STS)
      IF (STS.EQ.9) GOTO 700
      OP = 6
      WRITE(FGDIR,'(I2.2)') DIR
      CALL BTREE (FILE1,2,0,KEY,9,FGREC,120,STS)
      IF (STS.EQ.5) THEN
         INFOE(1) = 'Facility ID = '
         INFOE(1)(15:23) = FGID
         INFOE(2) = 'Facility ID already exists'
         INFOA(1) = 'Please check facility ID in subdirectory'
         CALL ERRMSG(99,ROUTIN,' ')
         GOTO 600
      ENDIF
      IF (STS.NE.0) THEN
         INFOE(1) = 'ERROR INSERTING FILE'
         CALL ERRMSG(STS,ROUTIN,'D..HGL')
         GOTO 990
```

```
      ENDIF
      GOTO 600
700   CONTINUE
* ==> close FACILITY.XDB in subdirectory
      CALL BTREE (FILE2,1,0,KEY,9,FGREC,120,STS)
      IF(STS.EQ.0) OPEND=.FALSE.
800   CONTINUE
      DIR = DIR + 1
      IF (DIR.GT.NEDIR) GOTO 990
      GOTO 500
990   CONTINUE
      IF(OPEND) CALL BTREE(FILE2,1,0,KEY,9,FGREC,120,STS)
      CALL BTREE (FILE1,1,0,KEY,9,FGREC,120,STS)
* ==> overwrite FACILITY.XDB
      CALL COMMAND('COPY FACTEMP.XDB FACILITY.XDB >NUL'C,ERR)
      CALL COMMAND('DEL FACTEMP.XDB >NUL'C,ERR)
999   END
* ==> subroutine for creating general facility information
      SUBROUTINE CRFAC(FILE1,SUCESS)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*2 FILE1(64),MAKE(32)
      LOGICAL SUCESS
$INCLUDE:'BTEROR.INC'
* ==> file information for FACILITY.XDB
      DATA MAKE/120,1024,3,5*0,1,9,0,5*0,11,4,2,5*0,49,7,3,5*0/
      SUCESS=.TRUE.
      ROUTIN = CRFAC
      CALL BTREE (FILE1,14,0,'FACTEMP.XDB',10,MAKE,64,STS)
      IF (STS.NE.0) THEN
         INFOE(1) = 'ERROR IN CREATE FILE'
         CALL ERRMSG(STS,ROUTIN,'X..987')
         SUCESS=.FALSE.
      ENDIF
      RETURN
      END
```

8 MANAGEMENT PROCEDURES

The success of the MRPM system depends greatly on the response the functional user receives to questions and problems reported to the operators/maintainers through the system telephone hotline. A toll-free 800 number should be provided to all users. This hotline should be used to report possible improvements, enhancements, questions, problems, and system failures. Top priority must be given to providing the user with a fast response to all problems and questions.

A three-part report log, shown in Figure 1, has been designed to assist in the management of the user requests. Each report log is assigned a unique sequential number for tracking purposes. The persons making and receiving the report are recorded with the message. The hotline operator will take immediate action to answer questions. All actions taken should be noted on the form. The function name should be kept when a specific function is being addressed. The red copy should be filed in the official logbook until the action is complete.

Many reports logs must be reviewed by a supervisor to determine what action must be performed, the priority of the log, and the person that must process the log. The white and yellow copies can be sent to the processor.

The processor should take all actions as soon as possible and record the action on the report log. All pertinent information should be given to the supervisor for review. The processor keeps the yellow copy as a record.

For quality assurance, the reviewer should analyze the entire command to ensure that it works completely. This is the most important step in ensuring a working system.

The problem log is returned to the hotline operator who will call the original reporter and relate the action taken. This step ensures the original user is kept informed. The completed white form replaces the red action form in the official report logbook.

The system supervisor should review the problem logs periodically to ensure that progress is being made on problem resolution. Written and oral feedback to all participants is required to maintain a good working environment.

A periodic newsletter should be mailed to each system user. The newsletter should include a short description of new features requested by the users. Include the user's name and organization, giving as much credit to the user as possible. Make sure that users know the system exists to support them. Give short summaries of changes to the system since the last newsletter.

Periodic system updates should be scheduled as needed by problem corrections and additions.

The most important part of the management process is training of system operators and maintainers. Both onsite and periodic Army-wide training courses should be offered for system users to be kept current.

Maintenance Resource Prediction Model

Report Log Number: _____

Report: Reporters Name/Org: _____ Tel. No.: _____

Date: _____ Time: _____ Received by: _____

Message: _____

_____

_____

_____

Action taken: _____

_____

Related commands: _____

Review: Reviewed by: _____ Date: _____

Priority: _____ Sent to: _____ Date sent: _____

Comments: _____

_____

_____

_____

_____

Process: Processed by: _____ Completion date: _____

Action taken: _____

_____

_____

_____

_____

Review: Reviewed by: _____ Date: _____

Comments: _____

_____

_____

Approval: _____

Feedback: Date reporter called: _____

Callers name: _____

_____

Figure 1. MRPM report log.

9 RESOURCES

Five basic functions must be maintained to provide full-service system support:

1. Supervision
2. Functional user training
3. Hotline
4. PC system maintenance
5. Newsletter.

Supervision

The supervisor's functions include scheduling training; review and assignment of report logs; and management of all corrections, improvements, and problem identification. This function will consume approximately 25 percent of a person year at a GS-11 level, which would be approximately $12,000/yr.

Functional User Training

There are three types of training:

1. Self-teach using the user's manual.
2. Onsite training of users.
3. Centralized training.

For the self-teach method, the user must have access to someone who can answer questions as the self-teaching progresses. This function is usually performed by the Hotline Operator.

Onsite training entails sending one person to a site for a minimum of 3 days. The installation provides equipment and the training room. The cost for each such session would be:

GS-11 trainer, 5 days @ $200/day	$1000
TDY, 4 days @ $100/day, plus air and car	1000
Supplies, manuals, etc.	300
Total	$2300

This is by far the best possible training method for both the trainer and the students. Five classes/yr at $2300 each would be $11,500.

Centralized training is the most expensive way for the Army to train. All students must travel to one central site. The central site must rent computer equipment that will probably not be the type used by any of the students at their installations. Training facilities must be obtained. A central Army training center must be paid to plan and conduct the training. During this training, there should be no more than two students assigned to one PC. Estimated costs would be:

GS-11 trainer, 5 days @ $200/day	$1000
TDY, 4 days @ $100/day, plus air and car	1000
Supplies, manuals, etc.	300
Computer rental (if available), $100/day @ 3 days	300 each
Room rental, $100/day @ 3 days	300
Student TDY, 4 days $100/day plus air	800 each
For a class of 20 students (10 computers)	Total $21,600

Hotline

The hotline is a telephone number that is used to support users by answering user questions, handling user problems, and accepting suggestions for improvement. This number is given to all users. The person answering the hotline should be able to either answer all basic problems, obtain the answer quickly, or refer the request to someone else for action. This activity would require about 15 percent of one GS-9 or $8000/yr.

PC System Maintenance

One standard system will be required at an initial purchase cost of $35,000. Annual system hardware maintenance costs would run $2000/yr. The U.S. Army Construction Engineering Research Laboratory (USACERL) will transfer the system to the U.S. Army Engineering and Housing Support Center (USAEHSC) at a cost of $20,000. Two GS-9 Fortran programmers must be trained on the use of the system. Training will take 6 months at a cost of $30,000. Normal annual requirement will be the equivalent of one half-time person at a cost of $15,000/yr.

Newsletter

A quarterly newsletter should inform users of updates and answers to common questions. The annual cost is estimated at $5000.

Cost Summary

Initial Costs	
USACERL transfer	$20K
Personal computer	$35K
Fortran program (2 at 6 mo)	$30K
Total	$85K
Annual Costs	
Supervisor (1/4 time)	$12K
User training - five classes onsite	$12K
Hotline (15 percent)	$8K
PC maintenance	$2K
Fortran programmer (1/2 time)	$15K
Newsletter	$5K
Basic supplies	$4K
Total	$58K